Е.В. Костина

READING FOR INFORMATION AND LEARNING

Обучение ознакомительному чтению

МЕТОДИЧЕСКОЕ ПОСОБИЕ

Иваново 2016

Министерство образования и науки
Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение
высшего образования
Ивановский государственный
химико-технологический университет

Е. В. Костина

READING FOR INFORMATION AND LEARNING

Обучение ознакомительному чтению

МЕТОДИЧЕСКОЕ ПОСОБИЕ

Иваново 2016

КОСТИНА Е.В.

READING FOR INFORMATION AND LEARNING

Обучение ознакомительному чтению: метод. пособие для студентов, обучающихся по специальности «Информационные системы и технологии»/ Е.В. Костина; Ивановский государственный химико-технологический университет.- Иваново, 2016. – 103 с.

Методическое пособие состоит из нескольких разделов и включает 25 аутентичных текстов по специальности «Информационные системы и технологии». Цель – научить студентов быстро извлекать информацию из текста. Для достижения поставленной цели разработана система упражнений. Материалы данного пособия помогают формировать навыки ознакомительного чтения, на основе сигнальных элементов, смысловых опор и фоновых знаний извлекать из текста необходимую информацию.

Для создания использовались ресурсы Интернет (материалы сайта Wikipedia), а также пособие Э.Д. Фролькис «Учись извлекать информацию при чтении» (Learning to read for information)/ Российская Академия наук, кафедра иностранных языков, Санкт-Петербург, 2000.

Может быть использовано студентами, аспирантами, изучающими английский язык.

Печатается по решению редакционно-издательского совета Ивановского государственного химико-технологического университета.

Рецензенты:
кандидат филологических наук А.Н. Смирнова (РЭУ им. Г.В. Плеханова, Ивановский филиал);
кандидат педагогических наук Л.А. Кабанова (Ивановская государственная сельскохозяйственная академия)

РЕКОМЕНДАЦИИ ПО РАБОТЕ С ПОСОБИЕМ

Данное пособие предназначено для обучения чтению научной литературы на английском языке по специальности «Информационные системы и технологии». Оно основано на материалах электронной энциклопедии, содержащей обширную универсальную информацию об информационных системах и их применении. Чтение как вид речевой деятельности направлено на извлечение информации из текста, на понимание его содержания. Однако степень полноты извлечения информации может быть разной в зависимости от цели чтения. Можно выделить несколько видов чтения исходя из коммуникативной установки: изучающее, ознакомительное, просмотровое, поисковое.

Изучающее чтение предполагает полное и точное понимание текста. *Ознакомительное чтение* направлено на выявление основного содержания текста. *Просмотровое чтение* имеет целью определить в общем виде тему и предмет изложения, степень новизны информации для того, чтобы понять, есть ли необходимость читать текст дальше. *Поисковое чтение* направлено на поиск необходимой конкретной информации. Из всех видов чтения ознакомительное – самое распространенное при чтении литературы по специальности как на родном, так и на иностранном языке.

В данном пособии овладение навыками и умениями ознакомительного чтения осуществляется в два этапа.

На первом этапе цель состоит в том, чтобы научить обучающихся ориентироваться в тексте по смыслу с опорой :

а) на грамматические и лексические элементы;

б)на логико-смысловые связи, выраженные словами-сигналами;

в) на композиционную структуру текста

г) на фоновые знания учащихся.

Такие опоры помогают найти нужную информацию.

На втором этапе закрепляются навыки ознакомительного чтения, и одновременно вырабатывается умение переносить сформированные навыки на узкоспециальные тексты.

Так как ознакомительное чтение, как правило, является неподготовленным чтением, то при обучении следует учитывать некоторые характерные особенности. Научный текст, как правило, обладает некоторой избыточностью в плане выражения для убеждения, доказательства, наглядности материала. Поэтому не весь материал представляет одинаковый интерес для читателя. В каждом тексте есть более информированные и менее информированные части. Таким образом, при обучении ознакомительному чтению задача состоит в том, чтобы научить их извлекать необходимую информацию, не отвлекаться на второстепенные детали следить за основной мыслью автора, отраженной в логической структуре текста.

Для оптимизации обучения ознакомительному чтению важно правильно подобрать материал для работы. Так как содержание науки информатики и технологии, связанные с производством компьютеров меняются очень быстро, трудно отследить все новшества и усовершенствования в этой области, и вносить изменения в содержание учебников по данному предмету. Поэтому предпочтение отдавалось аутентичным текстам – сообщениям описательного характера, охватывающим универсальные области изучаемого предмета. Подбирались не сложные по форме изложения тексты, с тем, чтобы учащийся мог сосредоточить внимание на понимании и извлечении информации.

С точки зрения содержания, отобранные тексты представляют собой законченное смысловое целое, достаточно информативны и интересны. Объем составляет 2000-5000 тысяч печатных знаков. Тексты снабжены заголовками. Тексты не расположены по степени трудности, так как не предполагается их последовательное прохождение. Перед текстами есть задания, формулирующие конкретную цель работы с данным текстом на уровне отдельных абзацев и текста в целом, даются рекомендации для ее достижения. Так как ознакомительное чтение не предполагает использование словаря, учащихся при чтении необходимо настроить на восприятие текста без словаря с опорой на знакомую лексику, на ключевые слова и лексику по специальности, на интернациональные слова, на так называемые сигнальные элементы текста, поддерживающие логико-смысловые связи в научном тексте и помогающие ориентироваться в структуре текста, а также фоновые знания предмета. Однако для того, чтобы упростить понимание особо сложных терминов и идиоматических выражений, после каждого текста дается их перевод, относящийся к конкретной ситуации в тексте. С помощью этих опор можно выделить основную и дополнительную информацию, существенные детали, а также заключение и выводы.Для смысловой ориентировки тексты разделены на абзацы, которые содержат, как правило, законченную мысль. Композиционная структура абзаца помогает понять его смысл, так как основная идея или информация чаще содержаться в начале и\или в конце абзаца. Развитие умения ознакомительного чтения также подразумевает использование учащимися фоновых знаний по предмету.

СТРУКТУРА ПОСОБИЯ

Пособие включает несколько разделов, каждый из которых представлен текстами заданиями. Раздел *“Learn to Read in Search Information*” – обучающий. Материалы раздела способствуют формированию навыков ознакомительного чтения и умению извлекать из текста информацию, опираясь на сигнальные элементы текста, смысловые опоры и фоновые знания. Каждый текст в этом разделе сопровождается комплексом заданий, помогающих извлечь информацию, перечень соответствующих сигнальных слов и конструкций. Первый текст каждой части сопровождается особенно подробными заданиями, которые помогают раскрыть содержание текста и могут служить образцом для последующей работы.

Второй раздел - *”Text for Training and Control”* - предназначен для закрепления умений ознакомительного чтения. В заданиях к текстам определяется только цель чтения. Задания этого раздела предусматривают контроль понимания прочитанного.

I. LEARN TO READ IN SEARCH INFORMATION

1. READING FOR THE MAIN INFORMATION (IDEA)

There are various ways of finding (idea) in the text.

Useful hints:

It is often helpful to look for the main information

1) by following the key word(s) together with the words related to it (them)
2) by concentrating on international words and familiar vocabulary
3) by finding the position of the key sentence which usually includes the key word and often appears at the beginning or\at the end of the paragraph
4) by using background knowledge of the subject.

TEXT 1

ASSIGNMENT 1.

Look through the following text. Pay attention to the key words and the words related to them: *to define, a computer, calculating, device, development, programmable.*

Formulate the main idea of the text.

Answer the question: *How was the meaning of the word "computer" changed with the passing of time?*

ASSIGNMENT 2.

Concentrate on the international words and words known to you which can be understood without dictionary. They are *in italic* in the text. These words are: *to define, computer, the term, to qualify, standards, calculating, the abacus, European mathematics, engineering, to be programmed, automatically, form of programmability, to conceptualize and design, mechanical computer, a number of technologies, analog computer, basis of computation, digital computer.*

ASSIGNMENT 3.

Look through paragraph 1. Concentrate on the beginning and the end.

Formulate the main idea.

Answer the question: *What was the origination of the earliest computing devices?*

ASSIGNMENT 4.

Look through paragraph 2. Concentrate on the beginning.

Formulate the main idea.

Answer the question: *What was changed in computing devices in the 19th century?*

ASSIGNMENT 5.

Look through paragraph 3. Concentrate on the beginning and the end.

Formulate the main idea.

Answer the question: *What kind of improvements in the 20th century allowed to design more modern computers?*

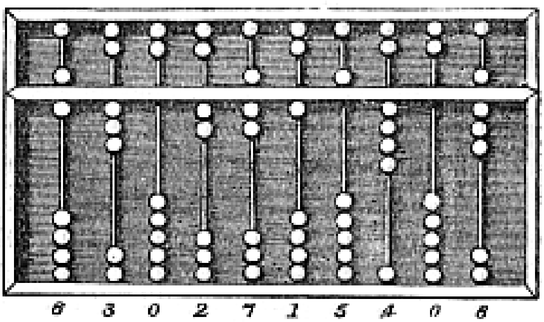
ASSIGNMENT 6.

Give a summary of the text.

History of Computing (Part I)

2351 t. un.

1. It is difficult *to define* any one device as the earliest *computer*. The very definition of a computer has changed and it is therefore impossible to identify the first computer. Many *devices* once called "computers" *would no longer qualify* as such by today's *standards*. Originally, *the term* "computer" referred to a person who performed *numerical calculations* (a human computer), often with the aid of *a mechanical calculating device*. Examples of early mechanical computing devices included *the abacus*, the slide rule and sometimes the astrolabe. The end of *the Middle Ages* saw a re-invigoration of *European mathematics* and



engineering, and Wilhelm Schickard's 1623 device was the first of a number of mechanical calculators constructed by European engineers.

2. However, none of those devices fit the modern definition of a computer because they *could not be programmed*. In 1801, Joseph Marie Jacquard *made an improvement* to the textile loom that used *a series of punched paper cards* as a template to allow his loom to weave intricate patterns *automatically*. The resulting Jacquard loom was an important step in the *development of computers* because the use of punched cards to define woven patterns can be viewed as an early *form of programmability*. In 1837, Charles Babbage was the first *to conceptualize and design a fully programmable mechanical computer* that he called "The Analytical Engine". Due to limited finance, and an inability to resist tinkering with the design, Babbage never actually built his Analytical Engine.

3. *Large-scale automated data processing* of punched cards was performed for the US Census in 1890 by *tabulating machines* designed by Herman Hollerith and manufactured by the Computing Tabulating Recording Corporation, which later became IBM. By the end of the 19th century *a number of technologies* that would later prove useful in the realization of practical computers had begun to appear: the punched card, boolean algebra, the vacuum tube (thermionic valve) and the teleprinter. During the first half of the 20th century, many *scientific computing needs* were met by increasingly *sophisticated analog computers*, which used *a direct mechanical or*

electrical model of the problem as *a basis for computation*. However, these were not *programmable* and generally lacked the versatility and accuracy of *modern digital computers*.

re-invigoration – воодушевление, *зд*. второе дыхание
versatility – универсальность
tinkering – лудить, паять
template – образец, шаблон

TEXT 2

ASSIGNMENT 1.

Look through the following text.

Follow the key words: *digital electronics, programmability, design, stored program architecture, technologies, computer, devices* together with the words related to them. Formulate the main idea of the text.

ASSIGNMENT 2.

Look trough the text more carefully so as to answer the questions:

1. What were the main changes in computing devices in the 20^{th} century?

2. Who made the design of modern computers?

3. What way do we use computers today?

ASSIGNMENT 3.

Give a summary of the text.

History of Computers (Part II)

3583 t. un.

1. A succession of steadily more powerful and flexible computing devices were constructed in the 1930s and 1940s, gradually adding the key features that are seen in modern computers. The use of *digital electronics* and *more flexible programmability* were vitally important steps, but defining one point along this road as "the first digital electronic computer" is difficult. Notable achievements include:

- Konrad Zuse's electromechanical "Z machines". The Z3 (1941) was the first working machine featuring *binary arithmetic,* including floating point arithmetic and a measure of programmability. In 1998 the Z3 was proved to be Turing complete, therefore being the world's first *operational computer*.
- The Atanasoff-Berry Computer (1941) which used *vacuum tube based computation*, *binary numbers*, and *regenerative capacitor memory*.
- The secret British Colossus computer (1944), which had *limited programmability* but demonstrated that a device using

thousands of tubes could be reasonably reliable and *electronically reprogrammable*.

- The Harvard Mark I (1944), a large-scale electromechanical computer with limited programmability.
- The US Army's Ballistics Research Laboratory ENIAC (1946), which used *decimal arithmetic* and was the first *general purpose electronic computer*.

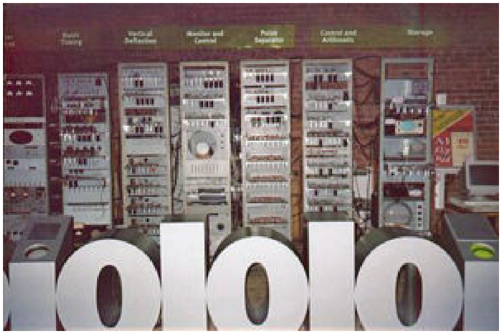
2. Several developers of ENIAC, recognizing its flaws, came up with a far more flexible and elegant *design*, which came to be

known as the *stored program architecture* or von Neumann architecture. This design was first formally described by John von Neumann in the paper "First Draft of a Report on the EDVAC", published in 1945. A number of projects to develop computers based on the stored program architecture commenced around this time, the first of these being completed in Great Britain. The first to be demonstrated working was the Manchester Small-Scale Experimental Machine (SSEM) or "Baby". However, the EDSAC, completed a year after SSEM, was perhaps the first practical implementation of the stored program design.

3. Nearly all modern computers implement some form of the stored program architecture, making it the single trait by which the word "computer" is now defined. By this standard, many earlier devices would no longer be called computers by today's definition, but are usually referred to as such in their *historical context*. While *the*

technologies used in computers have changed dramatically since the first electronic, general-purpose computers of the 1940s, most still use the von Neumann architecture. The design made the universal computer a *practical reality.*

4. *Vacuum tube-based computers* were in use throughout the 1950s, but were largely replaced in the 1960s by *transistor-based devices*, which were smaller, faster, cheaper, used less power and were more reliable. These factors allowed computers to be produced on an unprecedented *commercial scale*. By the 1970s, the adoption of *integrated circuit technology* and the subsequent creation of *microprocessors* such as the Intel 4004 caused another leap in size, speed, cost and reliability. By the 1980s, computers had become sufficiently small and cheap to replace simple *mechanical controls in domestic appliances* such as washing machines. Around the same time, computers became widely accessible for *personal use by individuals* in the form of home computers and the now ubiquitous personal computer. In conjunction with the widespread growth of *the Internet* since the 1990s, personal computers are becoming as common as the television and the telephone and almost all *modern electronic devices* contain a computer of some kind.



a flaw – ошибка, упущение

trait – особенность, характерная черта

ubiquitous – повсеместный

TEXT 3

ASSIGNMENT 1.

Look through paragraph 1 of the following text. Follow the key words *to be programmed, to be given, to store, to carry out, to execute, to add, to move, to send, to read, to tell computer, to jump ahead or backwards* and the word related to them.

Formulate the main idea of the paragraph.

Answer the question: *How can a computer be programmed?*

ASSIGNMENT 2.

Look through paragraph 2. Follow the key words *program execution, the flow of control.* Concentrate on the beginning and the end of the paragraph.

Formulate its main idea.

Answer the question: *What is the flow of control?*

ASSIGNMENT 3.

Look through paragraph 3. Pay attention to the key words *"think", to make decisions, "guessing", re-arranging, "big picture"* and the words related to them.

Formulate the main idea of the paragraph.

Answer the question: *Why can't a computer program be compared in some way with human brain?*

ASSIGNMENT 4.

Give a summary of the text.

Stored Program Architecture

3050 t. un.

1. The defining feature of modern computers which distinguishes them from all other machines is that they *can be programmed*. That is to say that *a list of instructions* (the program) *can be given* to the computer and it *will store* them and *carry them out* at some time in the future.

In most cases, computer instructions are simple: *add* one number to another, *move* some data from one location to another, *send* a message to some external device, etc. These instructions *are read* from the computer's memory and *are* generally *carried out (executed)* in the order they were given. However, there are usually specialized instructions *to tell the computer to jump ahead or backwards* to some other place in the program and *to carry on executing* from there. These are called *"jump" instructions (or branches)*. Furthermore, jump instructions may be made to happen conditionally so that *different sequences of instructions may be used* depending on the result of some previous calculation or some external event. Many computers directly support *subroutines* by providing a type of jump that "remembers" the location it jumped from and another instruction to return to that point.

2. *Program execution* might be likened to *reading a book*. While a person will normally read each word and line in sequence, they may at times *jump back* to an earlier place in the text or *skip*

sections that are not of interest. Similarly, a computer may sometimes *go back and repeat* the instructions in some section of the program over and over again until some internal condition is met. This is called *the flow of control* within the program and it is what allows the computer to *perform tasks* repeatedly without human intervention. Comparatively, person using a pocket calculator can perform a basic arithmetic operation such as adding two numbers with just a few button presses. But to add together all of the numbers from 1 to 1,000 would take thousands of button presses and a lot of time—with a near certainty of making a mistake. On the other hand, a computer may be programmed to do this with just a few simple instructions. Once told to run the program, the computer will perform the repetitive addition task without further human intervention. It will almost never make a mistake and a modern PC can complete the task in about a millionth of a second.

3. However, computers *cannot "think"* for themselves in the sense that they only *solve problems* in exactly the way they are *programmed to*. Many modern computers are able *to make some decisions* that *speed up* the execution of some programs by *"guessing"* about the outcomes of certain jump instructions and *re-arranging* the order of instructions slightly without changing their meaning. However, computers cannot intuitively *determine a more efficient way to perform the task* given to them because they do not have *an overall understanding* of what the task, or the *"big picture"*, is. In other words, a computer programmed to add up the numbers one by one would do exactly that without regard to efficiency or alternative solutions.

a branch- переход

a subroutine – подпрограмма

TEXT 4

ASSIGNMENT 1.

Look trough the following text. Find the main ideas, concentrating on the beginning and the end of the paragraph.

Answer the questions:

1. What is a computer program?

2. What is a computer error?

3. What is a machine code?

4. What is a fundamental concept of storing programs?

5. What are the main techniques in writing a computer program?

6. What is a compiler?

7. What is it used for?

8. Why is it difficult to develop a software system?

ASSIGNMENT 2.

Give a summary of the text.

Programs

4285 t. un.

1. In practical terms, a *computer program* might include anywhere from a dozen instructions to many millions of instructions

for something like a word processor or a web browser. *A typical modern computer can execute billions of instructions* every second and nearly never make a mistake over years of operation. Large computer programs may take teams of *computer programmers* years to write and the probability of the entire program having been written completely in the manner intended is unlikely. *Errors* in computer programs are called *bugs*. Sometimes bugs are *benign* and do not affect the usefulness of the program, in other cases they might cause the program to completely *fail (crash),* in yet other cases there may be *subtle problems.* Sometimes otherwise benign bugs may be used for *malicious intent*, creating a *security problems*. Bugs are usually not *the fault* of the computer. Since computers merely execute the instructions they are given, bugs are nearly always *the result of programmer error* or an oversight made in the program's design.

2. In most computers, *individual instructions* are stored as *machine code* with each instruction being given *a unique number* (its operation code or opcode for short). The simplest computers are able to perform any of a handful of different instructions, the more complex computers have several hundred to choose from—each with *a unique numerical code*. Since *the computer's memory* is able to store numbers, it can also store the instruction codes. This leads to the important fact that *entire programs* (which are just lists of instructions) can be represented as *lists of numbers* and can themselves be manipulated inside the computer just as if they were numeric data. *The fundamental concept* of storing programs in the computer's memory alongside the data they operate on is *the crux of the von Neumann*, or stored program, *architecture*. In some cases, a computer might store some or all of its program in memory that is

kept separate from the data it operates on. This is called *the Harvard architecture* after the Harvard Mark I computer. Modern von Neumann computers display some traits of the Harvard architecture in their designs, such as in *CPU caches*.

3. While it is possible to write computer programs as *long lists of numbers (machine language*) and this *technique* was used with many early computers, it is extremely *tedious* to do so in practice, especially for *complicated programs*. Instead, each basic instruction can be given *a short name* that is indicative of its function and easy to remember—*a mnemonic* such as ADD, SUB, MULT or JUMP. These mnemonics are collectively known as *a computer's assembly language*. Converting programs written in assembly language into something the computer can actually understand (machine language) is usually done by a computer program called an assembler. Machine languages and the assembly languages that represent them (collectively termed low-level programming languages) tend to be unique to a particular type of computer.

4. Though considerably easier than in machine language, writing long programs *in assembly language* is often *difficult and error prone*. Therefore, most *complicated programs* are written in more abstract *high-level programming languages* that are able to express the needs of the computer programmer more conveniently (and thereby help reduce programmer error). High level languages are usually *"compiled"* into machine language (or sometimes into assembly language and then into machine language) using another computer program called *a compiler*. Since high level languages are more abstract than assembly language, it is possible to use *different compilers* to translate the same high level language program into the

machine language of many different types of computer. This is part of the means by which *software* like video games may be made available for *different computer architectures* such as personal computers and various *consoles. The* task of *developing large software systems* is an *immense intellectual effort*. It has proven, historically, to be very difficult to produce software with an acceptably high reliability, on a predictable schedule and budget. The academic and professional discipline of *software engineering* concentrates specifically on this problem.

benign – неопасные
to be subtle – едва различим
malicious – злонамеренный, умышленный
oversight – недостаток, оплошность
alongside – наравне
the crux – загадка, главный вопрос
cache- кэш-память
to be prone – склонны

TEXT 5

ASSIGNMENT 1.

Look through paragraph 1 and 2 of the following text. Formulate the main ideas concentrating on the beginning and on the end of each paragraph.

Answer the questions:

1. What is a computer's memory?

2. How does it work?

3. What are RAM and ROM?

4. What way do RAM and ROM work?

5. What is flash memory?

ASSIGNMENT 2.

Read paragraph 3. Formulate its main idea. Find in the paragraph the examples of working I/O.

Answer the questions:

1. What are peripherals?

2. How do they work in different situations?

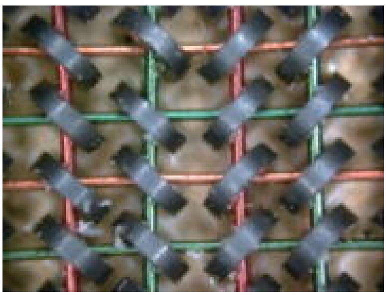
ASSIGNMENT 3.

Give a summary of the text.

Memory and Input/Output (I/O)

3454 t. un.

1. *A computer's memory* may be viewed as *a list of cells* into which numbers may be placed or read. Each cell has *a numbered "address"* and can store *a single number*. The computer may be instructed to "put the number 123 into the cell numbered 1357" or to "add the number that is in cell 1357 to the number that is in cell 2468 and put the answer into cell 1595". The information stored in memory may



represent practically anything. Letters, numbers, even computer instructions may be placed into memory with equal ease. In almost all modern computers, each *memory cell* is *set up to store binary numbers in groups of eight bits* (called *a byte)*. Each byte is able to represent 256 different numbers; either from 0 to 255 or -128 to +127. A computer may store any kind of information in memory as long as it can be somehow represented in *numerical form*. Modern computers have billions or even trillions of bytes of memory.

2. *Computer main memory* comes in *two principal varieties*: *random access memory* or RAM and *read-only memory* or ROM. RAM can be read and written to anytime the CPU commands it, but ROM is *pre-loaded* with data and software that never changes, so the CPU can only read from it. *ROM* is typically used *to store the computer's initial start-up instructions*. In general, *the contents of RAM* is erased when *the power to the computer is turned off* while ROM retains its data indefinitely. In a PC, the ROM contains *a specialized program* called the *BIOS* that orchestrates *loading the computer's operating system* from *the hard disk drive* into RAM whenever the computer *is turned on or reset.* In *embedded computers*, which frequently do not have disk drives, all of *the software* required to perform the task *may be stored in ROM*. Software that is stored in ROM is often called *firmware* because it is notionally more like hardware than software. *Flash memory* blurs the distinction between ROM and RAM by retaining data when turned off but being rewritable like RAM. However, flash memory is typically much slower than conventional ROM and RAM so its use *is restricted* to *applications* where high speeds are not required.

3. *I/O* is the means by which *a computer* receives *information* from the outside world and *sends results back*. Devices that provide input or output to the computer are called *peripherals*. On a typical personal computer, peripherals include *inputs* like *the keyboard and mouse*, and *outputs* such as *the display and printer*. *Hard disks, floppy disks and optical discs* serve as both inputs and outputs. *Computer networking* is another form of I/O. Practically any device that can be made to interface digitally may be used as I/O. Engine Control Unit of a modern automobile might read the position of the pedals and steering wheel, the output of the oxygen sensor and devices that monitor the speed of each wheel. The output devices include the various lights and gauges that the driver sees as well as the engine controls such as the spark ignition circuits and fuel injection systems. In a digital wristwatch, the computer reads the buttons and causes numbers and symbols to be shown on the liquid crystal display. Often, I/O devices are *complex computers* in their own right with their own CPU and memory. A graphics processing unit might contain fifty or more tiny computers that perform the calculations necessary to display 3D graphics. Modern desktop computers contain many smaller computers that assist the main CPU in performing I/O.

consecutive – последовательно, непрерывно

to blur – *зд.* стирать

gauges – измерительные приборы

notionaly – теоретически, символически

firmware – встроенные программы, «зашитые программы»

TEXT 6

ASSIGNMENT 1.

Look trough the following text. Pay attention to the key words: *supercomputer, software tools, highly-calculation intensive task, innovative design, parallel processing system, etc.* Say what this text is about.

ASSIGNMENT 2.

Concentrate on the international words, which can be understood without a dictionary and terms known to you. They are *in italic* in the text. These words are: *processing capacity, speed of calculations, fast scalar processor, a vector processor, the shelf units, highly – tuned computer clusters, commodity processor, source-based software solutions, ad hoc computer clusters, mainstream desktop operating system, programming language, complex detail engineering, memory hierarchy.*

ASSIGNMENT 3.

Look through paragraph 1. Formulate its main idea.

Answer the questions:

1. When did first supercomputer appear?

2. What are the basic principals of a supercomputer work?

ASSIGNMENT 4.

Look through paragraph 2. Concentrate on the details so as to answer the questions:

1. What kind of software do supercomputers use?

2. What are the main features of supercomputer software operating?

ASSIGNMENT 5.

Look through paragraph 3 and 4. Concentrate on the detail so as to answer the questions:

1. What are the main uses of supercomputers?

2. What are the peculiarities of supercomputer design?

ASSIGNMENT 6.

Give the summary of the text.

Supercomputer

***3523 t. un*.**

1. *A* ***supercomputer*** is a computer that leads the world in terms of *processing capacity*, particularly *speed of calculation*, at the time of its introduction. Supercomputers introduced in the 1960s were designed

primarily by *Seymour Cray* at Control Data Corporation (CDC). The term *supercomputer* itself is rather fluid, and today's supercomputer tends to become tomorrow's normal computer. CDC's early machines were simply *very fast scalar processors*, some ten times the speed of the fastest machines offered by other companies. In the 1970s most supercomputers were dedicated to *running a vector processor*. The early and mid-1980s saw *machines* with a modest number of vector processors *working in parallel* become the standard. Typical numbers of processors were in the range 4–16. In the later 1980s and 1990s, attention turned from *vector processors* to *massive parallel processing systems* with thousands of "ordinary" CPUs, some being *off the shelf units* and others being *custom designs*. Today, parallel designs are based on *"off the shelf" server-class microprocessors*, such as the PowerPC, IA-64, or x86-64, and most modern supercomputers are now *highly-tuned computer clusters* using *commodity processors* combined with *custom interconnects*.

2. *Software tools* for distributed processing include standard APIs such as MPI and PVM, and open *source-based software solutions* such as *Beowulf* and *openMosix* which facilitate the creation of a sort of *"virtual supercomputer"* from a collection of ordinary workstations or servers. *Technology* like *ZeroConf* pave the way for the creation of *ad hoc computer clusters*. An example of this is the distributed rendering function in *Apple's Shake compositing application*. Computers running the Shake software merely need to be in proximity to each other, in networking terms, to automatically discover and use each other's resources. While no one has yet built an *ad hoc computer cluster* that rivals even yesteryear's supercomputers, *the line* between desktop, or even laptop, and supercomputer is beginning *to blur*, and

is likely to continue to blur as *built-in support for parallelism* and *distributed processing* increases in *mainstream desktop operating systems*. An *easy programming language* for supercomputers remains an *open research topic* in Computer Science.

3. *Supercomputers* are used for *highly calculation-intensive tasks* such as problems involving *quantum mechanical physics*, *weather forecasting*, *climate research*, *molecular modeling* (computing the structures and properties of chemical compounds, biological macromolecules, polymers, and crystals), *physical simulations* (such as simulation of airplanes in wind tunnels, simulation of the detonation of nuclear weapons, and research into nuclear fusion), *cryptanalysis*, and the like. Major universities, military agencies and scientific research laboratories are heavy users.

4. *Supercomputers* using custom CPUs traditionally gained their speed over conventional computers through the use of *innovative designs* that allow them *to perform many tasks in parallel*, as well as *complex detail engineering*. They tend to be specialized for certain types of computation, usually *numerical calculations*, and perform poorly at more general computing tasks. Their *memory hierarchy* is very carefully designed to ensure the processor is kept fed with data and instructions at all times. Their *I/O systems* tend to be designed to support high *bandwidth*, with latency less of an issue, because supercomputers are not used for *transaction processing*.

fluid – размыт, неясен
modest – умеренный, ограниченный
to pave – *зд.* проложить дорогу, открыть путь

proximity – близость, схожесть

to rival – соперничать, конкурировать,

bandwidth – пропускная способность средств ввода\ вывода

latency – задержка

vector processor – матричный процессор

ad hoc*(лат.)* – после, *зд.* суперсовременные

API – интерфейс прикладного программирования

MPI – интерфейс передачи сообщений

PVM – параллельная виртуальная машина

Off the shelf – стандартные

Useful hints:

Text patterns may also aid in finding the main information (idea). Some texts are presented as Question-Answer patterns in which the author puts a question, then proceeds to answer it. The Question-Answer text pattern the main information (idea) is usually contained in the answer to the question stated in the text.

TEXT 7

ASSIGNMENT 1.

Look through the following text. Pay attention to the key words: *software, hardware, machine language, high-level programming* *language, system software, programming software, application software* together with there words related to them. Say what this text is about.

ASSIGNMENT 2.

Concentrate on the in international words and terms which can be understood without a dictionary. They are *in italics* in the text. These words are: *software, program, hardware, system software, word processor, application software, physical interconnections, devices, RAM, the central processing unit, machine language, binary values, processor instructions, a high-level programming language, compile, interpreter, assembly language, programming software, graphical user interface, the tools, an Integrated development environment.*

ASSIGNMENT 3.

Look through the paragraph 1. Concentrate on the question stated in the text.

Answer it using your background knowledge of the subject, key words and international words.

ASSIGNMENT 4.

Look through paragraph 2. Concentrate on the details so as to answer the questions:

1. Why is software called so?

2. Where is software kept in computers?

3. What is software and how does it work?

4. How do programming languages related to software?

ASSIGNMENT 4.

Look through paragraph 3. Concentrate on its key words and the beginning of each point.

Answer the questions:

1. What are main types of software?

2. What is the purpose of the system software?

3. What is programming software used for?

4. Where do we usually use application software?

ASSIGNMENT 5.

Give a summary of the text.

Computer Software

3366 t. un.

1. What is software and what do we use it for? *Software*, or *program*, enables a computer to perform *specific tasks*, as opposed to the physical components of the system (*hardware*). This includes *application software* such as a *word processor*, which enables a user to perform a task, and *system software* such as an *operating system*, which enables other software to run properly, by interfacing with hardware and with other software or custom software made to user specifications.

2. Computer software is so called in contrast to *computer hardware,* which encompasses the *physical interconnections* and *devices* required to store and execute (or run) the software. In computers, software is loaded into *RAM* and executed in *the central processing unit*. At the lowest level, software consists of *a machine language* specific to an individual processor. A *machine language* consists of groups of *binary values* signifying *processor instructions* (object code), which change the state of the computer from its preceding state. Software is an ordered sequence of instructions for changing the state of the computer hardware in a particular sequence. It is usually written in *high-level programming languages* that are easier and more efficient for humans to use (closer to natural language) than machine language. High-level languages *are compiled or interpreted* into machine language object code. Software may also be written in *an assembly language*, essentially, a mnemonic representation of a machine language using a natural language alphabet. Assembly language must be assembled into object code via an assembler.

3. The term "software" was first used in this sense by *John W. Tukey* in 1957. In computer science and software engineering, *computer software* is all computer programs. Practical computer systems divide software into *three major classes*: system software, programming software and application software.

- *System software* helps run the computer hardware and computer system. It includes operating systems, device drivers, diagnostic tools, servers, windowing systems, utilities and more. *The purpose* of systems software is *to insulate* the applications programmer as much as possible from the details of the particular computer complex being used, especially memory and other hardware features, and such accessory devices as communications, printers, readers, displays, keyboards, etc.

- *Programming software* usually *provides* tools *to assist* a programmer *in writing computer programs* and *software* using different programming languages in a more convenient way. *The tools* include text editors, compilers, interpreters, linkers, debuggers, and so on. *An Integrated development environment* (IDE) merges those tools into *a software bundle*, and a programmer may not need to type multiple commands for compiling, interpreter, debugging, tracing, and etc., because the IDE usually has an advanced *graphical user interface,* or GUI.
- *Application software* allows end users *to accomplish* one or more specific (non-computer related) tasks. *Typical applications* include industrial automation, business software, educational software, medical software, databases, and computer games. Businesses are probably the biggest users of application software, but almost every field of human activity now uses some form of application software. It is used to automate all sorts of functions.

a bundle - комплект, пакет

TEXT 8

ASSIGNMENT 1.

Look through the following text. Concentrate on the question. Answer it using your knowledge of the subject.

ASSIGNMENT 2.

Read the text and say if your answer agrees with this of the text. Answer the questions using key word that are *in italics* in the text:

1. How can we classify programmers?

2. What is the main task of a programmer?

3. How do programmers work?

4. What is the difference between application programmers and systems programmers?

ASSIGNMENT 3.

Give a summary of the text.

Programmer

3771 t. un.

1. What do we know about the people who make computer work? A *programmer* or *software developer* is someone who *programs* computers, that is, one who *writes computer software*. The term *computer programmer* can refer to a specialist in one area of computer programming or to a generalist who writes *code* for many kinds of software. One who *practices* or *professes a formal approach to programming* may also be known as a *programmer analyst*, *software*

engineer, computer scientist, or software analyst. A programmer's primary computer language (Java, C++, etc.) is often prefixed to the above titles, and those who work in a web environment often prefix their titles with *web*. Those proficient in computer programming skills may become famous, though this regard is normally limited to software engineering circles. Many of the most notable programmers are often labeled *hackers*.

2. Computer programmers *write, test, and maintain the detailed instructions*, called computer programs, that computers must follow to perform their functions. Programmers also conceive, design, and test logical structures for solving problems by computer. Many *technical innovations* in programming — *advanced computing technologies and sophisticated new languages and programming tools* — have redefined the role of a programmer and elevated much of the programming work done today. Job titles and descriptions may vary, depending on the organization.

3. Programmers write programs according to the *specifications* determined primarily by more senior programmers and by systems analysts. After the design process is complete, it is the job of the programmer *to convert* that design into *a logical series of instructions* that the computer can follow. The programmer codes these instructions in one of many *programming languages*. Different programming languages are used depending on *the purpose of the program*. Programmers generally know more than one programming

language and, because many languages are similar, they often can learn new languages relatively easily. In practice, programmers often are referred to by the language they know, e.g. as *Java programmers*, or by the type of function they perform or environment in which they work: for example, *database programmers*, *mainframe programmers*, or *Web developers*. Programmers test a program by running it to ensure that the program works as expected. If errors do occur, the programmer must make the appropriate change and recheck the program until it produces the correct results. This process is called *testing and debugging*. These are important parts of every programmer's job. Programmers may continue to fix these problems throughout the life of a program. Updating, repairing, modifying, and expanding existing programs sometimes called *maintenance programming*.

4. Computer programmers often are grouped into two broad types: *applications programmers* and *systems programmers*. Applications programmers write programs *to handle a specific job*, such as a program to track inventory within an organization. They also may revise existing packaged software or customize generic applications which are frequently purchased from independent software vendors. *Systems programmers*, in contrast, write programs *to maintain and control computer systems software*, such as operating systems and database management systems. These workers make changes in the instructions that determine how the network, workstations, and CPU of the system handle the various jobs they have been given and how they communicate with peripheral equipment such as printers and disk drives. Because of their knowledge of the entire computer system, systems programmers often help applications programmers debug, or determine the source of, problems that may occur with their programs.

2. FOLLOWING THE TRAIN OF THOUGHT

Useful hints:

Certain signal words and word combinations help to follow the train of thought thus facilitating the comprehension of the text. The following signals often serve

a) to introduce a new idea: *point out, show, indicate, demonstrate, emphasize + the key noun*

b) to specify a general statement: *in particular, particularly, especially, namely*

c) to support the previous statement: *indeed, really, in fact, as a matter of fact*

d) to express comparison or to contrast one thing to another: *as compared to, in comparison with, like, unlike, as ...as, than, not so ... as, instead, on the contrary, in the contrast to, but, however, yet, though, although, nevertheless, in spite of, despite, whereas, on the other hand, while, otherwise*

e) to express a summary or conclusion: *thus, accordingly, therefore, for the reason, so, as a result, in conclusion, concluding, summing up, summarizing, finally, at the end*

f) to introduce a general statements: *generally, in general, generally speaking*

g) to introduce additional information: *furthermore, further, moreover, in addition (to)*

h) to introduce illustrative examples: *for instance, for example, in other words.*

TEXT 9

ASSIGNMENT 1.

Look through the following text. Pay attention to the key words. These are: *central processing unit, microprocessor, transistor, integrated circuit.*

ASSIGNMENT 2.

Formulate the main idea of the text concentrating on its title, the beginnings of each paragraph and key words.

ASSIGNMENT 3.

Look through the text again. Concentrate on the international words or terms which can be understood without a dictionary. The words are *in italics* in the text. These words are: *central processing unit (CPU), digital computer, computer program instructions, processes data, the fundamental digital computer trait of programmability, single integrated circuit, microprocessor, a certain class of logic machines, the transistor, vacuum tubes*, *electrical relays, printed circuit boards, discrete (individual) components, a single semiconductor-based die*, *"small-scale integration" (SSI) devices.*

ASSIGNMENT 4.

Look through paragraph 1. Concentrate on the connectives between sentences, which may be considered as signals facilitating the comprehension of the text.

Find the statements in which

a) gives the definition of CPU. The signal *Generally* may be helpful,

b) the change in the thought is indicated. The signal *However* may be helpful.

Answer the question: *What is CPU?*

ASSIGNMENT 5.

Look through paragraph 2. Concentrate on the signals.

Find the statement introducing the example. The signal *in other words* may be helpful.

Answer the question: *What were the improvements in CPUs to make them smaller and more reliable?*

ASSIGNMENT 6.

Look through paragraph 3. Concentrate on the signals. Find the statement expressing the conclusion. The signal *thus* may be helpful.

Answer the question: *What was the influence of microprocessors on the development of computers?*

ASSIGNMENT 7.

Look through paragraph 4. Concentrate on the signals.

Find the statements

a) expressing the contrast. The signal *on the other hand* may be helpful,

b) expressing additional information. The signal *additionally* may be helpful.

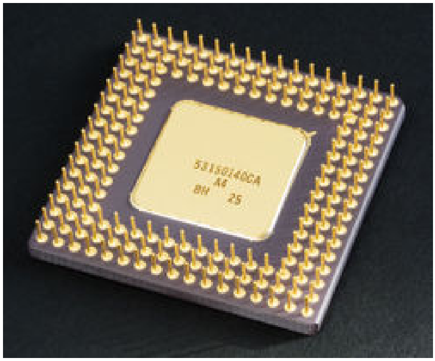
Answer the question: *What were the consequences of using microprocessors for computer?*

ASSIGNMENT 8.

Give the summary of the text.

Central Processing Unit

4027 t. un.



1. A *central processing unit (CPU),* or sometimes simply *processor*, is the component in a *digital computer* that interprets *computer program instructions* and *processes data.* CPUs provide *the fundamental digital computer trait of programmability*, and are one of the necessary components found in computers of any era, along with primary storage and input/output facilities. A CPU that is manufactured as *a single integrated circuit* is usually known as a *microprocessor*. Generally, the phrase "central processing unit" is a description of *a certain class of logic machines* that can execute computer programs. This broad definition can easily be applied to many early computers that existed long before the term "CPU" ever came into widespread usage. However, the term itself and its initialism have been in use in the computer industry at least since the early 1960s. The form, design and implementation of CPUs have changed dramatically since the earliest examples, but their fundamental operation has remained much the same.

2. The design complexity of CPUs increased as various technologies facilitated building smaller and more reliable electronic devices. The first such improvement came with *the advent of the transistor*. Transistorized CPUs during the 1950s and 1960s no longer had to be built out of bulky, unreliable, and fragile switching elements like *vacuum tubes* and *electrical relays*. In other words, with this improvement more complex and reliable CPUs could be built onto one or several *printed circuit boards* containing *discrete (individual) components*. During this period, a method of manufacturing many transistors in a compact space gained popularity. The *integrated circuit* (**IC**) allowed a large number of transistors to be manufactured on *a single semiconductor-based die*, or *"chip"*. At first only very basic non-specialized digital circuits such as NOR gates were miniaturized into ICs. CPUs based upon these "building block" ICs are generally referred to as *"small-scale integration" (SSI) devices*.

3. *The introduction of the microprocessor* in the 1970s significantly affected the design and implementation of CPUs. Since the introduction of the first microprocessor (the Intel 4004) in 1970 and the first widely used microprocessor (the Intel 8080) in 1974, this class of CPUs has almost completely overtaken all other central processing unit implementation methods. *Mainframe and minicomputer manufacturers* of the time launched proprietary IC development programs *to upgrade* their older computer architectures, and eventually produced *instruction set compatible microprocessors* that were *backward-compatible* with their older hardware and software. Thus, combined with the advent and eventual vast success of the now ubiquitous personal computer, the term "CPU" is now applied almost exclusively to microprocessors.

4. Previous *generations of CPUs* were implemented as *discrete components* and numerous *small integrated circuits (ICs)* on one or more circuit boards. Microprocessors, on the other hand, are CPUs manufactured on a very small number of ICs; usually just one. The overall smaller CPU size as a result of being implemented on a single die means faster switching time because of physical factors like decreased *gate parasitic capacitance*. This has allowed *synchronous microprocessors* to have clock rates ranging from tens of megahertz to several gigahertz. Additionally, as the ability to construct exceedingly small transistors on an IC has increased, the complexity and number of transistors in a single CPU has increased dramatically. This widely observed trend is described by *Moore's law,* which has proven to be a fairly accurate predictor of the growth of CPU (and other IC) complexity to date. While the complexity, size, construction, and general form of CPUs have changed dramatically over the past sixty years, it is notable that the basic design and function has not changed much at all. Almost all common CPUs today can be very accurately described as *von Neumann stored-program machines*.

an initialism - сокращение

an advent – появление, приход

bulky - громоздкий

a die - матрица

NOR gates – схема НЕ - ИЛИ

gate parasitic capacitance – входная паразитная емкость

TEXT 10

ASSIGNMENT 1.

Look through the following text. Outline the main ideas, concentrating on the heading and the beginning of each paragraph.

ASSIGNMENT 2.

Look through the text once more. Find

a) the statements indicating the change in the train of thought. The signal *however* may be helpful

b) the statement expressing the conclusion. The signal *although* may be helpful.

In paragraph 3 try to guess the meaning of the word *crippleware* from the context.

ASSIGNMENT 3.

Answer the questions:

1. What was shareware created for?

2. What are the differences and common features of free/open source software and shareware?

3. What kind of problems have the users got nowadays?

ASSIGNMENT 4.

Give a summary of the text.

Shareware

4011 t. un.

1. *Shareware* is a *marketing method* for *computer software*. Shareware software is typically obtained *free of charge*, either by *downloading from the Internet* or *on magazine cover-disks*. A user tries out the program, and thus shareware has also been known as "*try before you buy"*. A shareware program is accompanied by *a request for payment*, and *the software's distribution license* often requires such a payment. The term *shareware* was coined by Bob Wallace to describe his word processor PC-Write in the mid-1980s. Wallace came up with the name that stuck, but many consider the "fathers" of the shareware marketing model to be Jim "Button" Knopf and Andrew Fluegelman. Their coordinated offerings of PC-File (database) and PC-Talk (telecommunications) programs, respectively, pre-dated PC-Write by several months. Button referred to his distribution method as "*user supported software",* and Fluegelman called his *freeware*. Among the three of them, they clearly established shareware as a *viable software marketing method.* Via the shareware model, PC-File and PC-Talk made Button and Fluegelman millionaires.

2. *Free/open source software and shareware* are similar in that they can be obtained and used *without monetary cost*. Usually shareware differs from free/open source software in that *requests of voluntary shareware fees* are made, often within the program itself, and in that *source code* for shareware programs is generally not available in a form that would allow others to extend the program. Notwithstanding that tradition, some free/open source software

authors ask for voluntary donations, although there is no requirement to do so. Free/open source software is usually *compatible with the strict ASP shareware guidelines*. *Original shareware* programs were applications *running under DOS*, but are now more commonly utilities running on Microsoft Windows, although gaming, editing and other examples also exist. There is *a technical difference* between *shareware* and *demos*. Up to the early 1990s, shareware could easily be upgraded to the full version by adding the other episodes or full portion of the game; this would leave the existing shareware files intact. *Demos* are different in that they are *self-contained programs* which are *not* upgradeable to the full version. A good example is the *Descent* shareware versus the *Descent II* demo; players were able to retain their saved games on the former but not the latter.

3. In the 1980s and early-to-mid 1990s shareware was considered to be a concept for *independent software writers* to receive a degree of renumeration for their labor. However, after that the shareware model began *to degrade* as the term was used by commercial startups offering commercial software and *labeling non-functional or limited demo versions* (known as crippleware) as shareware. As a result, the term shareware has shown reduced usage in recent years, replaced by either demo for trial software or freeware for full editions. However, it must be stressed that the shareware software is not always so limited in function, as demonstrated with programs such as The Bat!, GetRight, WinZip, and WinRAR. One problem is *the lack* of a *solid definition*. Some shareware groups have limited definitions, allowing '*nag screens*' that remind the user to buy the software and refuse to accept any software with limited functionality, such as demos, trial use, or crippled software. Most groups, such as the Association of

Shareware Professionals, the Software Industry Professionals group and PC Shareware clearly state their position that any software marketed as ‘*try before you buy’* is shareware. Another issue is the high percentage of *commercial failures*. A very large percentage of shareware projects are commercial failures. Active projects commonly see less than 0.5% of downloaders convert to paying customers, and projects may be *victims of software piracy*, dropping sales by half again. It is argued that many projects could become successful by following some simple business practices.

‘nag screens’ – *зд.* рекламные окна, заставки
victims – жертвы

TEXT 11

ASSIGNMENT 1.

Look trough the following text. Outline its main idea, concentrating on the information in the beginning of every paragraph and key words *database, information, database models, data* and words related to them.

Answer the questions:

1. What is database?

2. What do we use database for?

ASSIGNMENT 2.

Look through paragraph 1 and 2.

Find the statements

a) specifying general statement. Use the signal *typically*.

Answer the question: *What is a schema?*

b) where new ideas are introduced. Use the signals *the earliest, the first*.

Answer the question: *What was Bachman's aim?*

c) which introduce contrast. Use the signals *however, but*.

Answer the question: *What is a relational model?*

ASSIGNMENT 3.

Look trough paragraph 3. Concentrate on the comparison of different types of database. The repetition of the same word combination may be helpful.

Answer the question: *How did the idea of using database develop during the 1980s*?

ASSIGNMENT 4.

Look through paragraph 4. Find the statements expressing conclusion. Use the signal *so*, *so that, at last*.

ASSIGNMENT 5.

Give a summary of the text.

Database

4853 t. un.

1. The term or expression *database* originated within the computer industry. A possible definition is that a database is *a collection of records* or *information* which is stored in a computer in a *systematic (i.e. structured) way*, so that a computer program can consult it to answer questions. The items retrieved in answer to queries become information that can be used to make decisions. *The computer program used to manage and query a database* is known as a *database management system* (DBMS). The properties and design of database systems are included in the study of information science. *The central concept of a database* is that of a *collection of records*, or *pieces of knowledge*. Typically, for a given database, there is *a structural description of the type of facts* held in that database: this description is known as *a schema*. The schema describes the objects that are represented in the database, and the relationships among them. There are a number of different ways of organizing a schema, that is, of *modeling the database structure*: these are known as *database models* (or data models). The term *database* refers to the collection of related records, and the software should be referred to as the *database management system* or DBMS. When the context is unambiguous, however, many database administrators and programmers use the term *database* to cover both meanings.

2. The earliest known use of the term '*data base'* was in June 1963, when the System Development Corporation sponsored a symposium under the title *Development and Management of a Computer-centered Data Base*. *Database* as a single word became

common in Europe in the early 1970s and by the end of the decade it was being used in major American newspapers. The first database management systems were developed in the 1960s. A pioneer in the field was *Charles Bachman*. Bachman's early papers show that his *aim* was *to make more effective use of the new direct access storage devices* becoming available. *The relational model* was proposed by E. F. Codd in 1970. He criticized existing models for confusing *the abstract description of information structure* with *descriptions of physical access mechanisms*. For a long while, however, the relational model remained of academic interest only. While CODASYL systems and IMS were conceived as practical engineering solutions taking account of the technology as it existed at the time, the relational model took a much more theoretical perspective, arguing (correctly) that hardware and software technology would catch up in time. Among the first implementations were *Michael Stonebraker's Ingres* at Berkeley, and *the System R project* at IBM. Both of these were *research prototypes*, announced during 1976. The first commercial products, *Oracle* and *DB2,* did not appear until around 1980. The first successful database product for microcomputers was dBASE for the CP/M and PC-DOS/MS-DOS operating systems.

3. During the 1980s, *research activity* focused on distributed database systems and database machines, but these developments had little effect on the market. Another important *theoretical idea* was the *Functional Data Model,* but apart from some specialized applications in genetics, molecular biology, and fraud investigation, the world took little notice. In the 1990s, attention shifted to *object-oriented databases*. These had some success in fields where it was necessary *to handle more complex data* than relational systems could easily cope with, such as spatial databases, engineering data, and multimedia data.

Some of these ideas were adopted by the relational vendors, who integrated new features into their products as a result. In the 2000s, the fashionable area for innovation is the *XML database*. As with object databases, this has spawned a new collection of startup companies, but at the same time the key ideas are being integrated into the established relational products. XML databases aim *to remove the traditional divide between documents and data*, allowing all of an organization's information resources to be held in one place, whether they are highly structured or not.

4. So databases are used in many applications, spanning virtually the entire range of computer software. Databases are the preferred method of storage for *large multiuser applications*, where *coordination between many users* is needed. Even individual users find them convenient, though, and many electronic mail programs and personal organizers are based on standard database technology. Software database drivers are available for most database platforms so that application software can use a common application programming interface (API) to retrieve the information stored in a database. Two commonly used database APIs are JDBC and ODBC. At last a database is also a place where you can store data and then arrange that data easily and efficiently.

to query – запрашивать информацию
unambiguous – недвусмысленный, точно выраженный
to conceive – полагать, задумывать
fraud – мошенничество, обман
spatial – пространственный
to span – охватывать, измерять

3. READING FOR IMPORTANT SUBPOINTS

Useful hints:

Important subpoints of the text are often introduced by such signal words, word combinations and special marks, such as:

- *first, second, third;*
- *in the first place, in the second place, finally;*
- *in one sense, in another sense;*
- *on the one hand, on the other hand;*

(a), (b), (c): (1), (2), (3).

TEXT 12

ASSIGNMENT 1.

Look through the following text. Pay attention to its heading and the key words *integrated circuit* and words related to them.

Formulate the main idea of the text.

ASSIGNMENT 2.

Concentrate on the international words and terms related to them which can be understood without a dictionary. They are *in italics* in the text. These words are: *a monolithic integrated circuit, a miniaturized electronic circuit, a thin substrate of semiconductor material, semiconductor devices, transistors, discrete electronic components, a precursor idea, small ceramic squares, bidimentional* *or tridimentional compact grid, principle of p-n junction isolation, a key concept, circuit boards, a microprocessor, wafer-scale integration, defect-free manufacturability problems.*

ASSIGNMENT 3.

Look through paragraph 1. Concentrate on the beginning and the end of the paragraph.

Answer the question: *How did integrated circuits allow to improve the work of computers?*

ASSIGNMENT 4.

Look through paragraph 2. Concentrate on the subpoints *first, then* and names and dates so as to show the history of invention of integrated circuit.

ASSIGNMENT 5.

Look through paragraphs 3 and 4. Find the important subpoints using the signals *first, next, further, final*.

Answer the questions:

1. Who first used integrated circuits?

2. What makes it attractive to use of integrated circuits in producing computers?

3. Why is it impossible to produce even more complex IC?

ASSIGNMENT 6.

Concentrate on the end of the text. Find the statement where two ideas are contrast. The signal *but* may be helpful.

Answer the question: *What components may be put on the SOC?*

ASSIGNMENT 7.

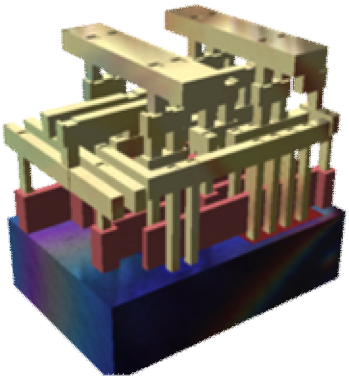
Give a summary of the text.

Integrated Circuit

5082 t. un.

1. *A monolithic integrated circuit* (also known as IC, microcircuit, microchip, silicon chip, computer chip or chip) is *a miniaturized electronic circuit* (consisting mainly of semiconductor devices, as well as passive components) that has been manufactured in the surface of *a thin substrate of semiconductor material*. Integrated circuits were made possible by experimental discoveries which showed that *semiconductor devices* could perform the functions of *vacuum tubes,* and by mid-20th-century *technology advancements in semiconductor device fabrication*. The integration of large numbers of *tiny transistors* into a small chip was an enormous improvement over the manual assembly of circuits using *discrete electronic components*. The integrated circuit's mass production *capability, reliability, and building-block approach* to circuit design ensured the rapid adoption of standardized ICs in place of designs using discrete transistors.

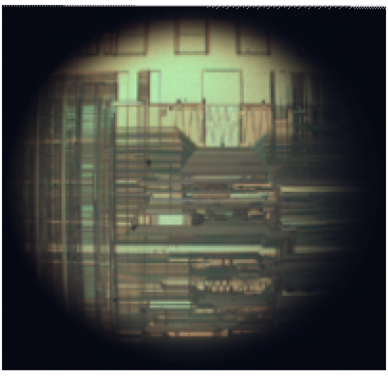
2. The integrated circuit was first conceived by a radar scientist, *Geoffrey W.A. Dummer* (born 1909), working for the *Royal Radar Establishment* of the British Ministry of Defence, and published in



Washington, D.C. on May 7, 1952. Dummer unsuccessfully attempted to build such a circuit in 1956. *A precursor idea* to the IC was to create *small ceramic squares (wafers)*, each one containing a single miniaturized component. Components could then be integrated and wired into *a bidimentional or tridimentional compact grid*. This idea, which looked very promising in 1957, was proposed to the US Army by *Jack Kilby*, and led to the short-lived *Micromodule Program.* However, as the project was gaining momentum, Kilby came up with a new, revolutionary design: the IC. The first integrated circuits were manufactured independently by two scientists: Jack Kilby of Texas Instruments filed a patent for a "Solid Circuit" made of germanium on February 6, 1959. *Robert Noyce* of Fairchild Semiconductor was awarded a patent for a more complex *"unitary circuit"* made of Silicon on April 25, 1961. Noyce credited Kurt Lehovec of Sprague Electric for the *principle of p-n junction isolation* caused by the action of a biased p-n junction (the diode) as *a key concept* behind the IC.

3. The first integrated circuits contained only a few transistors. Called *"Small-Scale Integration" (SSI)*, they used circuits containing transistors numbering in the tens. SSI circuits were crucial to early *aerospace projects*, and vice-versa. Both the Minuteman missile and Apollo program needed *lightweight digital computers* for their *inertially-guided flight computers*. These programs purchased almost all of the available integrated circuits from 1960 through 1963. The

next step in the development of integrated circuits, taken in the late 1960s, introduced devices which contained hundreds of transistors on each chip, called "*Medium-Scale Integration*" (*MSI*). They were attractive economically because while they cost little more to produce than SSI devices, they allowed more complex systems to be produced using smaller *circuit boards*, less assembly work (because of fewer separate components), and a number of other advantages. Further development, driven by the same economic factors, led to "*Large-Scale Integration*"(*LSI*) in the mid 1970s, with tens of thousands of transistors per chip. The final step in the development process, starting in the 1980s and continuing on, was *"Very Large-Scale Integration" (VLSI),* with hundreds of thousands of transistors, and beyond (well past several million in the latest stages). For the first time it became possible to fabricate a CPU on a single integrated circuit, to create *a microprocessor*. In 1986 the first *one megabit RAM chips* were introduced, which contained more than one million transistors. Microprocessor chips produced in 1994 contained more than three million transistors.



4. To reflect further growth of the complexity, the term *ULSI* that stands for "*Ultra-Large Scale Integration*" was proposed for *chips of complexity more than 1 million of transistors*. However, there is no *qualitative leap* between VLSI and ULSI, hence normally in technical texts the "VLSI" term covers ULSI as well, and "ULSI" is reserved only for cases when it is necessary to emphasize the chip

complexity, e.g. in marketing. The most extreme integration technique is *wafer-scale integration (WSI),* which uses *whole uncut wafers containing entire computers*. Attempts to take this step commercially in the 1980s failed, mostly because of *defect-free manufacturability problems*, and it does not now seem to be a high priority for industry. The WSI technique failed commercially, but advances in semiconductor manufacturing allowed for another attack on IC complexity, known as *System-on-Chip (SOC)* design. In this approach, components traditionally manufactured as *separate chips to be wired together on a printed circuit board* are designed to occupy a single chip that contains memory, microprocessor(s), peripheral interfaces, Input/Output logic control, data converters, and other components, together composing the whole electronic system.

to conceive - представлять
precursor - предшествующая
grid - решетка
p-n junction – p-n переход
biased – смещение
wafer – плата

TEXT 13

ASSIGNMENT 1.

Look through paragraph 1 of the following text. Follow the key words *Intel Corporation* and proper names together with the words related to them. Outline the main points contained in the paragraph.

Answer the question: *What kind of equipment does the company produce?*

ASSIGNMENT 2.

Look through paragraph 2. Find its main points concentrating on the first statement. Find the important subpoints using the signals. The words *at its beginning, then, by the end, after, at last* and dates may be helpful.

Answer the questions:

1. What were the main products of Intel first?

2. How did the business of Intel develop during the 1970s – 1980s?

3. What changes happened in the beginning of 2000s?

4. What is the position of Intel now?

ASSIGNMENT 3.

Give a summary of the text.

Intel Corporation

2955 t. un.

1. *Intel Corporation* is the world's largest *semiconductor company* and *the inventor of the x86 series of microprocessors*, the processor found in many personal computers. Intel was founded in 1968 by *Gordon E. Moore* (a chemist and physicist) and *Robert Noyce* (a physicist and co-inventor of the integrated circuit) when they left Fairchild Semiconductor. A number of other Fairchild employees also went on to participate in other Silicon Valley companies. Intel's fourth employee was *Andy Grove* (a chemical engineer), who ran the company through much of the 1980s and the high-growth 1990s. Grove is now remembered as the company's key business and strategic leader. By the end of the 1990s, Intel was one of the largest and most successful businesses in the world, though fierce competition within the semiconductor industry has since diminished its position.

2. Intel has grown through several distinct phases. *At its beginning*, Intel was distinguished simply by its ability to make *semiconductors*, and its primary product were *static random access memory (SRAM) chips*. Then Intel's business grew during the 1970s as it expanded and improved its manufacturing processes and produced a wider range of products, still dominated by various memory devices. While Intel created the first *microprocessor* in 1971, by the early 1980s its business was dominated by Dynamic random access memory chips. However, increased competition from *Japanese semiconductor manufacturers* had by 1983 dramatically reduced the profitability of this market, and *the sudden success of the IBM personal computer* convinced then-

CEO Grove *to shift the company's focus to microprocessors* and to change fundamental aspects of that business model. By the end of the 1980s this decision had proven successful, and Intel embarked on a 10-year period of unprecedented growth as the primary (and most profitable) *hardware supplier to the PC industry*. After 2000, growth in demand for high-end microprocessors slowed and competitors garnered significant market share, initially in *low-end and mid-range processors* but ultimately across the product range, and Intel's dominant position was reduced. In the early 2000s then-CEO Craig Barrett attempted *to diversify the company's business beyond semiconductors*, but few of these activities were ultimately successful. At last in 2005, CEO Paul Otellini reorganized the company to refocus its core processor and chipset business on platforms (enterprise, digital home, digital health, and mobility) which led to the hiring of over 10,000 new employees. But in September of 2006 due to falling profits, the company announced a restructuring that resulted in a layoff of 10,500 employees or about 10 percent of its workforce by July of 2007. In September 2006, Intel had nearly 100,000 employees and 200 facilities world wide. Its 2005 revenues were $38.8 billion and its Fortune 500 ranking was 49th. Its stock symbol is INTC, listed on the NASDAQ.

***fierce* -** жестокая

to garner – копиться, запасать

layoff – увольнение из-за остановки производства

CEO – *сокр.* исполнительный директор

TEXT 14

ASSIGNMENT 1.

Look through paragraph 1 of the following text. Outline the main point contained in the paragraph.

Answer the question: *What kind of equipment does IBM produce?*

ASSIGNMENT 2.

Look through paragraph 2. Find its main idea concentration on the first statement.

Find the important subpoints using the signals.The words *one, another, third* may be helpful.

Answer the question: *Why the words "Big Blue" are the nick of the company?*

ASSIGNMENT 3.

Look through paragraphs 3 and 4. Find its main ideas concentrating on the beginning of each paragraph.

Answer the questions:

1. What activities does IBM undertake except producing computer equipment?

2. What business activities does IBM undertake to improve its position in the market?

ASSIGNMENT 4.

Give a summary of the text.

IBM

2707 t. un.

1. *International Business Machines Corporation* (known as IBM or "*Big Blue")* is a *multinational computer technology corporation* headquartered in Armonk, New York, USA. The company is one of the few information technology companies with a continuous history dating back to the 19th century. IBM manufactures and sells *computer hardware, software*, *infrastructure services, hosting services and consulting services* in areas ranging from *mainframe computers to nanotechnology*. IBM's PC division was bought by *Chinese company Lenovo* on May 1, 2005. The company which became IBM was founded in 1889 as *Herman Hollerith and the Tabulating Machine Company.* It was incorporated as *Computing Tabulating Recording Corporation* (CTR)) on June 15, 1911, and was listed on the New York Stock Exchange in 1916. IBM adopted its current name in 1924, where it became a fortune 500 company.

2. *Big Blue* is *a nickname* for IBM; several theories exist regarding its origin. One theory, substantiated by people who worked for IBM at the time, is that *IBM field reps* coined the term in the 1960s, referring to the color of the mainframes IBM installed in the 1960s and early 1970s. "*All blue"* was a term used *to describe a loyal IBM*

customer, and business writers later picked up the term. Another theory suggests that Big Blue simply refers to *the Company's logo*. A third theory suggests that *Big Blue* refers to *a former company dress code* that required many IBM employees *to wear blue suits*. IBM has often been described as having *a sales-centric or a sales-oriented business culture.* Traditionally, many IBM executives and general managers are chosen from the sales force. Middle and top management are often enlisted to give direct support to salesmen when pitching sales to important customers.

3. IBM has been influenced by *the open source movement*, and began supporting *Linux* in 1998. The company invests billions of dollars in services and software based on Linux through the *IBM Linux Technology Center*, which includes over 300 Linux kernel developers. IBM has also released *code under different open-source licenses*, such as *the platform-independent software framework Eclipse and the Java-based relational database management system (RDBMS) Apache Derby.*

4. *The IBM Project Management Center of Excellence* (PM COE) is *a program* dedicated to *defining and executing* the steps IBM must take *to strengthen its project management capabilities*. Functioning as *IBM's think tank*, the PM COE combines external industry trends and directions with IBM business, organizational, and geographic requirements and insight. Upon this foundation deliverables (such as project management policy, practices, methods, and tools) are developed.

IBM field reps – *зд.* сотрудники

Coin – изобретать, придумывать

Useful hints:

The repetition of the same constructions, the same forms of pronouns, nouns, verbs, prepositions or their substitutes in the text is also a good help in revealing important subpoints.

TEXT 15

ASSIGNMENT 1.

Look through the following text. Follow the key words. These are: *Boolean algebra, algebra of two values, operations, logic* and the words related to them. Outline the main ideas contained in the text.

ASSIGNMENT 2.

Concentrate on the international words and terms which can be understood without a dictionary. They are *in italics* in the text. These words are: *Boolean algebra, the finitary algebra of two values, the algebra of real numbers, the logical operations , conjunction, disjunction, complement, Boolean domain of two values, calculus, digital logic, computer programming, and mathematical logic*, *machine code*, *assembly language, symbols, bit vectors, the numeric operations, core differentiating feature, a fundamental tool.*

ASSIGNMENT 3.

Look through paragraph 1.

Find the statements which

a) express contrast. The signal *but* may be helpful

b) basic principals of Boolen algebra are explained. The repetition of the same word combinations may be used as a signal. These are: *two values, finitary*.

ASSIGNMENT 4.

Look through paragraph 2.

Find the important subpoints related to the main idea.

The repetition of the same word combination through the text may be helpful.

Answer the question: *How do the principals of Bollen algebra work in digital logic?*

ASSIGNMENT 5.

Look through the paragraph 3 more carefully so as to be able to answer the question: *What way does Bollen algebra work in programming?*

ASSIGNMENT 6.

Give a summary of the text.

Boolean Algebra

2787 t. un.

1. *Boolean algebra* is *the finitary algebra of two values*. It resembles *the algebra of real numbers* as taught in high school, but with the numeric operations of multiplication xy, addition $x + y$, and negation $\neg x$ replaced by *the logical operations of conjunction* $x \wedge y$, *disjunction* $x \vee y$, and *complement* $\neg x$. The Boolean operations are these and all other operations obtainable from them by composition; equivalently, *the finitary operations on the set {0,1}*. Boolean algebra is the algebra of two values. These are usually taken to be 0 and 1, as we shall do here, although ⊥ and T, false and true, etc. are also in common use. For the purpose of understanding Boolean algebra any *Boolean domain of two values* will do.

2. Boolean algebra as *the calculus of two values* is fundamental to *digital logic, computer programming, and mathematical logic*, and is also used in other areas of mathematics such as *set theory and statistics*. Digital logic codes its symbols in various ways: as voltages on wires in high-speed circuits and capacitive storage devices, as orientations of a magnetic domain in ferromagnetic storage devices, as holes in punched cards or paper tape, and so on. Now it is possible to code more than two symbols in any given medium. For example one might use respectively 0, 1, 2, and 3 volts to code a four-symbol alphabet on a wire, or holes of different sizes in a punched card. In practice however the tight constraints of high speed, small size, and low power combine to make noise a major factor. This makes it hard to distinguish between symbols when there are many of them at a single site. Rather than attempting to distinguish between four voltages on one wire, digital designers have settled on two voltages

per wire, high and low. To obtain four symbols one uses two wires, and so on.

3. *Programmers* programming in *machine code*, *assembly language*, and other programming languages that expose the low-level digital structure of the data registers operate on whatever *symbols* were chosen for the hardware, invariably *bit vectors in modern computers* for the above reasons. Such languages support both *the numeric operations of addition, multiplication*, etc. performed on words interpreted as integers, as well as *the logical operations of disjunction, conjunction, etc.* performed bit-wise on words interpreted as bit vectors. Programmers therefore have the option of working in and applying the laws of either *numeric algebra or Boolean algebra* as needed. *A core differentiating feature* is carry propagation with the former but not the latter. Algebra being *a fundamental tool* in any area amenable to mathematical treatment, these considerations combine to make the algebra of two values of fundamental importance to computer hardware, mathematical logic, and set theory.

conjunction – соединение, сцепление
complement – дополнение, совокупность
domain – область определения
set theory – теория множеств
constraint – *зд.* ограничение
propagation – воспроизведение
amenable – подверженная
integer – целое число
bit-wise – поразрядный побитовый
register- запись, перечень

4. READING FOR DETAILS

Useful hints:

Details are often introduced by names, dates, figures, geographical names, etc.

TEXT 16

ASIGNMENT 1.

Look through the following text. Pay attention to the key words. They are: *computing hardware, devices, programmability, analog and digital computers*. Formulate the main idea of the text.

ASSIGNMENT 2.

Concentrate on the international words and terms, proper names and dates which can be understood without a dictionary. These words are: *the process of calculation and data storage, mechanical calculator, techniques, the binary numeral system, decimal system, to control the machine, modern computers, Schickard, Pascal, Leibniz, Jaquard, Babbage, Hollerith* and others.

ASSIGNMENT 3.

Look through paragraph 1. What do the italicized pronouns ***it*** mean in the text.

Concentrate on the details and answer the question: *What way did the humanity help itself in calculations?*

ASSIGNMENT 4.

Look through paragraph 2. Concentrate on the names so as to answer the questions:

1. Who created the first programmable computing device?

2. Who created the first general-purpose programmable computer?

ASSIGNMENT 5.

Look through paragraph 3. Find its main information concentrating on the words in italic and dates so as to answer the questions:

1. When were analog computers most popular?

2. What techniques were used in analog computers?

3. When did the era of modern computers begin? Why?

ASSIGNMENT 6.

Give a summary of the text.

History of Computing Hardware (Part I)

4108 t. un.

1. ***Computing hardware*** has been an important component of *the process of calculation* and *data storage* since ***it*** became useful for numerical values to be processed and shared. Humanity has used devices to aid in computation for millennia. One example is a device for establishing equality by weight: *the classic scales*. A more

arithmetic-oriented machine is *the abacus*. One of the earliest machines of this type was the Chinese abacus. In 1623 *Wilhelm Schickard* built the first *mechanical calculator* and thus became the father of the *computing era*. Since his machine used *techniques* such as *cogs and gears* first developed for clocks, it was also called a 'calculating clock'. Machines by *Blaise Pascal* (the Pascaline, 1642) and *Gottfried Wilhelm von Leibniz* (1671) followed. Around 1820, *Charles Xavier Thomas* created the first successful, mass-produced mechanical calculator, *the Thomas Arithmometer*, that could add, subtract, multiply, and divide. ***It*** was mainly based on Leibniz's work. Leibniz also described *the binary numeral system*, a central ingredient of all modern computers. However, up to the 1940s, many subsequent designs were based on the harder-to-implement *decimal system*. *John Napier* noted that multiplication and division of numbers can be performed by addition and subtraction, respectively, of *logarithms* of those numbers. *Slide rules* were used by generations of engineers and other mathematically inclined professional workers, until the invention of the pocket calculator.

2. In 1801, *Joseph-Marie Jacquard* developed a loom in which the pattern being woven was controlled by *punched cards*. The series of cards could be changed without changing the mechanical design of the loom. This was a *landmark* point in *programmability*. *The defining feature* of a "universal computer" is *programmability*, which allows the computer to emulate any other calculating machine by *changing a stored sequence of instructions*. In 1833, *Charles Babbage* moved on from developing his *difference engine* to developing a more complete design, *the analytical engine*, which would draw directly on Jacquard's punch cards for its programming. In 1835 Charles

Babbage described his analytical engine. It was the plan of a *general-purpose programmable computer*, employing punch cards for input and a steam engine for power. His initial idea was to use punch-cards *to control a machine* that could calculate and print logarithmic tables with huge precision (a specific purpose machine). Babbage's idea soon developed into a general-purpose programmable computer, his analytical engine. In 1890, *the United States Census Bureau* used punch cards and sorting machines designed by *Herman Hollerith,* to handle the flood of data from the decennial census mandated by the Constitution. Hollerith's company eventually became the core of IBM.

3. Before World War II, *mechanical and electrical analog computers* were considered the 'state of the art', and many thought they were the future of computing. *Analog computers* use continuously varying amounts of *physical quantities*, such as *voltages or currents*, or the *rotational speed of shafts*, to represent the quantities being processed. Unlike modern *digital computers,* analog computers are not very *flexible*, and need to be *reconfigured* (i.e., reprogrammed) manually to switch them from working on one problem to another. The art of analog computing reached its zenith with *the differential analyzer*, invented by *Vannevar Bush* in 1930. The era of *modern computing* began with a flurry of development before and during World War II, as *electronic circuits, relays, capacitors, and vacuum tubes* replaced mechanical equivalents and digital calculations replaced analog calculations. Machines such as the *Atanasoff-Berry Computer*, *the Z3*, *the Colossus*, and *ENIAC* were built by hand using circuits containing *relays or valves* (vacuum tubes), and often used *punched cards* or punched paper tape for input and as the main (non-volatile) *storage medium*. By 1954, *magnetic*

core memory was rapidly displacing most other forms of *temporary storage*, and dominated the field through the mid-1970s.

cogs and gears – шестерни и приводы
to incline - наклонять
precision - точность
decennial census – перепись (населения)
rotational speed of shafts – частота вращения осей
a flurry – шквал, вспышка
capacitor - конденсатор
valve – электронная лампа
volatile – переменная, изменчивая (среда)

TEXT 17

ASSIGNMENT 1.

Look through the following text. Outline the main idea of each paragraph, concentrating on the key words *in italic* and dates.

ASSIGNMENT 2.

Look through the text more carefully.

Concentrate on the details so as to answer the questions:

1. What were the main features of Z3?
2. What was the name of the first high –level programming language?
3. Why do we consider the ABC the first modern computer?
4. What were the main features of the Harvard Mark I computer?

5. What were the main features of the Colossus computer?

6. Why do we consider ENIAC the first electronic general-purpose computer?

7. What were its advantages?

ASSIGNMENT 3.

Give a summary of the text.

History of Computing Hardware (Part II)

3586 t. un.

1. Working in isolation in Germany, *Konrad Zuse* started construction in 1936 of his first *Z-series calculators featuring memory and (initially limited) programmability*. Zuse's purely mechanical, but already binary *Z1*, finished in 1938, never worked reliably due to problems with the precision of parts. Zuse's subsequent machine, the *Z3*, was finished in 1941. It was based on *telephone relays* and did work satisfactorily. The Z3 thus became the first *functional program-controlled computer*. Programs were fed into Z3 on *punched films*. In two 1936 patent applications, Konrad Zuse also anticipated that *machine instructions* could be stored in the same storage used for *data* - the key insight of what became known as *the Von Neumann architecture*. Zuse also claimed to have designed the first *higher-level programming language*, (Plankalkül), in 1945.

2. In 1938 *John Vincent Atanasoff and Clifford E. Berry of Iowa State University* developed *the Atanasoff-Berry Computer (ABC)*, *a special purpose electronic computer* for solving systems of linear equations. Though the ABC machine was not programmable, it was the *first modern computer* in several other respects, including the first to use *binary math and electronic circuits*.

3. In 1939, development began at *IBM's Endicott laboratories* on the *Harvard Mark I.* Known officially as the Automatic Sequence Controlled Calculator, the Mark I was *a general purpose electro-mechanical computer* built with IBM financing and with assistance from IBM personnel, under the direction of Harvard mathematician Howard Aiken. Its *design* was influenced by Babbage's Analytical Engine, using *decimal arithmetic and storage wheels and rotary switches* in addition to electromagnetic relays. It was *programmable* via *punched paper tape,* and contained several *calculation units* working in parallel. The Mark I was moved to Harvard University and began operation in May 1944.

4. The *Mk I Colossus* was built in 1942 by *Tommy Flowers* and his colleagues at the Post Office Research Station at Dollis Hill in London. Colossus was the first *totally electronic computing device*. The Colossus used a large number of *valves (vacuum tubes).* It had *paper-tape input* and was capable of *being configured* to perform a variety of *boolean logical operations* on its data.

5. The US-built *ENIAC (Electronic Numerical Integrator and Computer),* often called the *first electronic general-purpose computer*, publicly validated the use of electronics for large-scale computing. This was *crucial* for the development of *modern computing*, initially because of the enormous *speed advantage*, but ultimately because of

the potential for *miniaturization*. Built under the direction of *John Mauchly and J. Presper Eckert*, it was 1,000 times faster than its contemporaries. ENIAC's development and construction lasted from 1943 to full operation at the end of 1945. *A "program"* on the ENIAC was defined by the states of its *patch cables and switches,* a far cry from the stored program electronic machines that evolved from it. At the time, *unaided calculation* was seen as enough of a triumph to view the *solution of a single problem* as the object of a program. Adapting ideas developed by Eckert and Mauchly after recognizing the limitations of ENIAC, *John von Neumann* wrote a widely-circulated report describing a *computer design* in which *the programs and working data* were both stored in a *single, unified store*. *This basic design*, which became known as the *von Neumann architecture*, would serve as the basis for the development of the first really flexible, *general-purpose digital computers*.

An insight - понимание
To validate – утверждать, подтверждать
A far cry – большая разница
Ultimately – в конечном счете

TEXT 18

ASSIGNMENT 1.

Look through the following text. Outline the main points concentrating on the beginning and the end of each paragraph, the key words *Hewlett-Packard Company, a manufacturer of test and measurement instruments, an optical disc recording technology, distinguishing feature, extreme accuracy and stability, multifunction products* and the words related to them.

ASSIGNMENT 2.

Look through the text for details so as to be able to answer the questions:

1. When was the company founded?

2. Why was it named Hewlett-Packard?

3. What were its main products in the beginning?

4. What was a new technology developed by Hewlett-Packard?

5. What did the company focus on?

6. Why did the company earn the respect of users?

7. What were the main products if the company in the 1990s?

ASSIGNMENT 3.

Give a summary of the text.

The Hewlett-Packard Company

3535 t. un.

1. The *Hewlett-Packard Company*, commonly known as *HP*, is currently the world's largest *information technology corporation* and is known worldwide for its *printers* and *personal computers*. HP was founded on January 1, 1939 as *a manufacturer of test and measurement instruments*, by *Bill Hewlett* and *Dave Packard*. They had both graduated from Stanford University in 1934. The company originated in a garage in nearby Palo Alto while they were post-grad students at Stanford during the Great Depression. After he won a coin toss to name the company, Bill named it Hewlett-Packard. The company name, Hewlett-Packard, was derived from the founders' last names. Had Bill not won a coin toss, the company today might be known as Packard-Hewlett.

2. Their first product was *a precision audio oscillator, the Model 200A*. Their innovation was the use of *a small night-light bulb* as *a temperature dependent resistor* in a critical portion of the circuit. This allowed them to sell the Model

200A for $54.40 when competitors were selling less stable oscillators for over $200. One of the company's earliest customers was *The Walt Disney Company*, who bought eight Model 200B oscillators (at $71.50 each) for use in certifying *the Fantasound surround sound systems* installed in theaters for the movie *Fantasia*. *LightScribe* was invented by Hewlett-Packard engineer Daryl Anderson. LightScribe is *an optical disc recording technology* that utilizes specially *coated recordable CD and DVD media* to produce *laser-etched labels*. *The purpose* of LightScribe is to allow users to create *direct-to-disc labels* (as opposed to stick-on labels), using their optical disc writer.

3. The company was originally rather unfocused, working on a wide range of electronic products for industry and even agriculture. Eventually they elected to focus on *high-quality electronic test and measurement equipment*. Throughout the 1940s to well into the 1990s the company focused on making *signal generators, voltmeters, oscilloscopes, counters, and other test equipment.* Their *distinguishing feature* was pushing *the limits* of measurement range and accuracy. They also focused on *extreme accuracy and stability*, leading to a wide range of very *accurate, precise*, and *stable frequency counters, voltmeters, thermometers, and time standards*.

The company earned global respect for a variety of products. They introduced the world's first *handheld scientific electronic calculator* in 1972 (the HP-35), the first *handheld programmable* in 1974 (the HP-65), the first *alphanumeric, programmable, expandable* in 1979 (the HP-41C), and *the first symbolic and graphing calculator HP-28C*. HP entered the computer market in 1966 with the *HP 2100 / HP 1000 series of minicomputers*. A simple *accumulator-based design*, with *registers arranged* somewhat similarly to the *Intel x86 architecture* still used today, it would last 20 years and several attempts to replace it. It would give birth to the *HP 9800 and HP 250 series of desktop and business computers*, which predated the PCs by nearly a decade. In 1984, HP introduced both *inkjet and laser printers* for the desktop. Along with its scanner product line, these have later been developed into successful *multifunction products*, the most significant being *single-unit printer/scanner/copier/fax machines*. HP develops the hardware, firmware, and software that convert data into dots for the mechanism to print. In 1987, the Palo Alto garage where Hewlett and Packard started their business was designated as a California State historical landmark.

Useful hints:

Details may be revealed through comparison.
The text pattern in which various items are compared makes it easy to identify point's similarity, deference and contrast. The following signal words and word combinations denote:

1) **similarity**: *the same, identical to (with), equal to, similar to, like, undifferentiated, as ... as, both, both ... and, neither ... nor, have in common, be alike, as well as;*
2) **difference**: *different from, unlike, not so ... as, vary, differ in (from), be unlike to, be distinguished from, than, one ... the other,* (if two items are compared), *difference between, distinction;*
3) **contrast**: *while, whereas, but, however, in contrast to.*

TEXT 19

ASSIGNMENT 1.

Look through the following text. Concentrate on the beginning.

Answer the question: *Who, when and how founded the Apple Company?*

ASSIGNMENT 2.

a) find the statements indicating the train of thought.

The signals *despite, while, another, as well as, but* may be helpful.

Answer the questions:

1. Why were first computers of the Apple so successful?

2. What was another key to success for Apple?

b) concentrate on the signal *nevertheless.*

Answer the questions:

1. Why did Apple fail with their computers in 1983?

2. Who were the main competitors of Apple in the market?

3. What was the strategy of Apple in the early 1980s?

ASSIGNMENT 3.

Give a summary of the text.

The Apple Company

3032 t. un.

1. *Apple* was founded on April 1, 1976 by *Steve Jobs, Steve Wozniak and Ronald Wayne* to sell the *Apple I personal computer kit*.

They were hand-built in a garage of Jobs' parents, and the Apple I was first shown to the public at the *Homebrew Computer Club*. Eventually 200 computers were built. The Apple I was sold as a motherboard (with CPU, RAM, and basic textual-video chips) — not what is today considered a complete personal computer. *The Apple II* was introduced on April 16, 1977 at the first *West Coast Computer Faire*. Despite a price higher than competitors, it quickly pulled away from its two main rivals, the *TRS-80* and *Commodore PET*, to become the market leader in the late 70s due to its *color graphics, high build quality*, and *open architecture*. While early models used *ordinary cassette tapes* as storage devices, this was quickly superseded by the introduction of a *5 1/4 inch floppy disk drive* and *interface,* the *Disk II*. Another key to success for Apple was software. The Apple II was chosen by programmers *Dan Bricklin* and *Bob Frankston* to be the desktop platform for the first "killer app" of the business world—*the VisiCalc spreadsheet program*. VisiCalc created a business market for the Apple II, and the corporate market attracted many more software and hardware developers to the machine, as well as giving home users an additional reason to buy one—compatibility with the office.

2. By the end of the 1970s, Jobs and his partners had a staff of computer designers and a production line. The *Apple II* was succeeded by the *Apple III* in May 1980 as the company struggled to compete against *IBM* and *Microsoft* in the lucrative business and corporate computing market. The designers of the Apple III were forced to comply with Jobs' request to omit *the cooling fan*, and this ultimately resulted in thousands of recalled units due to *overheating*. An *updated version* was introduced in 1983, but it was also *a failure* due to bad press and wary buyers. Nevertheless, the principals of the company persevered with further innovations and marketing. Apple's sustained growth during the early 1980s was mostly due to its leadership in the *education sector* because of their *adaption of the programming language LOGO,* which was used in many schools with the *Apple II*. The drive into education was accentuated in *California* with the donation of one *Apple II* and one *Apple LOGO software package* to each public school in the state. The deal concluded between *Steve Jobs* and *Jim Baroux of LCSI*, and having required the support of *Sacramento*, established a strong and pervasive presence for Apple in all schools throughout California. The initial conquest of education environments was critical to Apple's acceptance in the home where the earliest purchases of computers by parents was in support of children's continued learning experience. Based on the marketing and technical savvy of Steve Jobs and Steve Wozniak, and the business expertise of Mike Markkula, Apple dominated the personal computer industry from 1977 to 1983.

lucrative – прибыльный, рентабельный
to comply – выполнять, осуществлять
to omit - пренебрегать
wary - осторожный
pervasive - распространяющееся
savvy – сообразительность

TEXT 20

ASSIGNMENT 1.

Look through the following text. Follow the key words *Microsoft Company, Windows, security* together with the words related to them. Give the main facts expressed about Microsoft.

ASSIGNMENT 2.

Look through the text more carefully for details. The signals *first, the early, finally, as well as* may be helpful.

Answer the questions:

1. What were main routes of Microsoft?
2. Why were versions of Windows so popular?
3. What was the main problem of Microsoft?
4. How did Microsoft solve this problem?

ASSIGNMENT 3.

Find the statements

a) expressing difference. The signal *unlike* may be helpful
b) expressing contrast. The signals *however, but* may be helpful.

ASSIGNMENT 4.

Give a summary of the text.

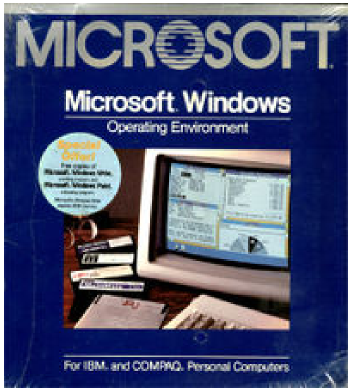
Microsoft Windows

3506 t. un.

1. *Microsoft Windows* is the name of *several families of proprietary software operating systems* by Microsoft. Microsoft first introduced an *operating environment* named *Windows* in November 1985 as an *add-on to MS-DOS* in response to the growing trend of *graphical user interfaces* (GUI) popularized by the Macintosh. *Microsoft Windows* eventually came to dominate the world's personal computer market. Microsoft has taken *two parallel routes in operating systems. One route* has been *the home user* and *the other* has been *the professional IT user*. *The dual route* has generally led to *the home versions with more "eye candy*" and *less functionality* in networking and security, and *professional versions with less "eye candy"* and better networking and security.

2. *The early versions of Windows* were often thought of as just *graphical user interfaces* or *desktops*, mostly because they were started from MS-DOS and used for file system services. However, even *the earliest 16-bit Windows versions* already assumed many *typical operating system functions,* notably having their own *executable file format* and providing their own *device drivers* (timer, graphics, printer, mouse, keyboard and sound) for applications. Unlike MS-DOS, Windows allowed users to execute *multiple graphical applications* at the same time, through cooperative multitasking. Finally, Windows implemented an *elaborate, segment-based, software*

virtual memory scheme which allowed it to run applications larger than available memory: *code segments and resources* were swapped in and thrown away when memory became scarce, and *data segments* moved in memory when a given application had relinquished processor control, typically waiting for user input.



3. *Security* has been *a hot topic* with Windows for many years, and even Microsoft itself has been *the victim of security breaches*. Due in some part to the widespread usage of Windows on personal computers, as well as a number of technical reasons there is reportedly *a fivefold greater amount of malware for Windows* than other operating systems such as *GNU/Linux, Unix, Mac OS X, and FreeBSD*. Windows was originally designed for *ease-of-use on a single-user PC* without a network connection, and did not have *security features* built in from the outset. Windows NT and its successors are designed for security (including on a network) and multi-user PCs, but was not designed for *Internet security* in mind as much since, when it was first developed, the Internet was less important. Combined with *flawed code* (such as buffer overflows), Windows is *a frequent target of worms and virus writers*. Furthermore, until Windows Server 2003 most versions of even Windows NT were shipped with important security features disabled by default, and vulnerable (albeit useful) system services enabled by default.

4. *Microsoft publicly* admitted their *ongoing security problems* shortly after the turn of the century and now *claims to regard security*

as their number one priority. The much-needed Automatic Update came first with Windows Me. As a result, *Service Pack 2* for Windows XP, as well as *Windows Server 2003*, was installed by users more quickly than it otherwise might have been. Microsoft releases *security patches* through its Windows Update service approximately once a month, although critical updates are made available at shorter intervals when necessary. In *Windows 2000* (SP3 and later), *Windows XP* and *Windows Server 2003*, updates can be automatically downloaded and installed if the user selects to do so.

add-on – добавление, приставка
trend - тенденция
eventually – в конечном счете, со временем
elaborate – тщательно продуманный, улучшенный
swap in – загружать, подкачивать
scarce – недостаточный, редкий
relinquish – оставлять, ослаблять
outset - начало
flawed code – поврежденный код
vulnerable - уязвимый
default – значение по умолчанию
security patches – обновление программ по безопасности
breach – брешь, дырка

II. TEXTS FOR TRAINING AND CONTROL

TEXT 21

ASSIGNMENT 1.

Look through the following text. Find its main information concentrating on the beginning of each paragraph.

ASSIGNMENT 2.

Look through the text one more. Concentrate on the signals so as to be able to answer the questions:

1. What was the first name of the company?

2. How did the company succeed in selling computers?

3. What are the main peculiarities of market strategy the company relies on?

4. What is Dell's market policy?

ASSIGNMENT 3.

Give a summary of the text.

Dell Inc.

3515 t. un.

1. *Dell Inc.*, an *American computer-hardware company* based in Round Rock, Texas, *develops, manufactures, sells and supports* a wide range of *personal computers,*

servers, data storage devices, network switches, personal digital assistants (PDAs), software, computer peripherals, and more. In 2006, *Fortune magazine* ranked Dell as No. 8 on its annual list of the *most-admired companies* in the United States. One publication has identified Dell as *one of 38 high-performance companies* in the S&P 500 which consistently *out-performed the market over the previous 15 years*. *Michael Dell*, while still a student at the University of Texas at Austin in 1984, founded the company as *PC's Limited* with just $1000. From Michael Dell's off-campus dorm room at Dobie Center, the startup aimed to sell IBM-compatible computers built from stock components.

2. In 1985, the company produced *the first computer in China of its own design* (the "Turbo PC"), which contained an *Intel 8088-compatible processor* running at a speed of 8 MHz. It advertised the systems in national computer magazines for sale directly to consumers, and custom-assembled each ordered unit according to a selection of options. Although not the first company to use this model, *PC's Limited* became one of the first to succeed with it. Eventually, Michael Dell dropped out of school to run the business full-time. The company grossed more than $6 million in its first year. In 1987, PC's Limited set up its *first on-site-service programs* in order *to compensate for the lack of local retailers* prepared to act as service centers. In 1988, *Dell's market capitalization* grew by $30

million to $80 million on its initial public offering day. The company changed its name to "*Dell Computer Corporation*" in 1988.

3. *The corporation markets specific brand* names to different consumer segments. It typically sells *the OptiPlex, Latitude, and Precision* names to medium-sized and large business customers, where the company's advertising emphasizes *long life-cycles, reliability and serviceability*. The *Dimension, Inspiron, and XPS brands* have an orientation towards *consumers, students, and small home office environments*, emphasizing *value, performance and expandability*. Dell recently re-introduced *the Dell XPS brand* to target *the lucrative gaming market*. *XPS desktop systems* use silver rather than the black cases found on newer Dell PCs. Dell has also diversified its product line to include peripheral products such as *USB keydrives, LCD televisions, and printers.*

4. Dell sells all its products both to *end-use consumers* and to *corporate customers*, using a *direct-sales model* via the Internet and the telephone network. Dell maintains a negative cash conversion cycle through use of this model: in other words, Dell Inc. receives payment for the products before it has to pay for the materials. Dell also practices *just-in-time (JIT) inventory management*, profiting from its attendant benefits. *Dell's JIT approach* utilizes *the "pull" system* by building computers only after customers place orders and by requesting materials from suppliers as needed. Since the original dominance of telephone ordering, the Internet has significantly enhanced Dell's business model, making it easier for customers and potential customers to contact Dell directly. Other computer manufacturers, including Gateway and Hewlett-

Packard, have attempted to adapt this same business model, but due to timing and/or retail-channel pressures they have not achieved the same results as Dell.

to gross – *зд.* увеличила общую сумму дохода
a retailer – розничный торговец
lucrative – прибыльный, рентабельный
attendant – сопровождающий, сопутствующий

TEXT 22

ASSIGNMENT 1.

Look through the following text. Follow the key words and their substitutes together with the words related to them. Find main information of each paragraph.

ASSIGNMENT 2.

Look through paragraph 1. Concentrate on the signals.

Answer the question: *Why did UNIX become so popular?*

ASSIGNMENT 3.

Look though paragraph 2.

Answer the question: *What are the main components of the UNIX system?*

ASSIGNMENT 4.

Look through paragraph 3.

Answer the question: *What is the aim of GNU?*

ASSIGNMENT 5.

Give a summary of the text.

Unix

3275 t. un.

1. *Unix* is *a computer operating system* originally developed in the 1960s and 1970s by a group of AT&T employees at Bell Labs including *Ken Thompson, Dennis Ritchie and Douglas McIlroy*. *Unix operating systems* are widely used in both *servers and workstations*. Both Unix and the C programming language were developed by AT&T and distributed to government and academic institutions, causing both to be ported to *a wider variety of machine families* than any other operating system. As a result, *Unix* became *synonymous with "open systems"*. Unix was designed to be *portable, multi-tasking and multi-user in a time-sharing configuration.* Under Unix, *the "operating system"* consists of many of utilities along with *the master control program, the kernel. The kernel* provides services *to start and stop programs, handle the file system and other common "high level" tasks* that most programs share, and, perhaps most importantly, *schedules access to hardware to avoid conflicts* if two programs try to access the same resource or device simultaneously. To mediate such

access, the kernel was given special rights on the system and led to the division between *user-space* and *kernel-space*.

2. The Unix system is composed of several components that are normally packaged together. By including — in addition to the kernel of an operating system — the development environment, libraries, documents, and the portable, modifiable source-code for all of these components, *Unix* was *a self-contained software system*. This was one of *the key reasons* it emerged into *an important teaching and learning tool* and had such a broad influence. The names and filesystem locations of the Unix components has changed substantially across the history of the system. Nonetheless, the V7 implementation is considered by many to have the canonical early structure: *Kernel* — originally found in /usr/sys, and composed of several sub-components. *Development Environment* — most implementations of Unix contained a development environment sufficient to recreate the system from source code. *Commands* — most Unix implementations make little distinction between commands (user-level programs) for system operation and maintenance, commands of general utility, and more general-purpose applications such as the text formatting and typesetting package. *Documentation* — Unix was the first operating system to include all of its documentation online in machine-readable form.

3. In 1983, *Richard Stallman* announced *the GNU project*, an ambitious effort to create *a free software Unix-like system; "free"* in that everyone who received a copy would be *free to use, study, modify, and redistribute it*. GNU's goal was achieved in 1992. Its own kernel development project, *GNU Hurd*, had not produced *a working kernel*, but *a compatible kernel* called *Linux* was released as free

software in 1992 under *the GNU General Public License*. The combination of the two is frequently referred to simply as "Linux", although the Free Software Foundation and some Linux distributions, such as *Debian GNU/Linux*, use the combined term *GNU/Linux.* Linux distributions, comprising Linux and large collections of compatible software have become popular both with hobbyists and in business.

to mediate – обеспечить, создать

to emerge – появляться

TEXT 23

ASSIGNMENT 1.

Look through the following text. Find its main information concentrating on paragraph 1, the beginning and the end of paragraph 2 and the last paragraph.

Using the key words and the signals answer the questions:

1. What are the basic principals of hacker subculture?

2. Where were the first hacker groups founded?

3. What way is the hackerdom divided? 4. What traditions does the hacker subculture develop?

ASSIGNMENT 2.

Give a summary of the text.

Hacker (academia)

3033 t. un.

1. In academia, *a hacker* is a person who *follows a spirit of playful cleverness and enjoys programming*. *The context of academic hackers* forms *a voluntary subculture* termed *the academic hacking culture*. Before communications between computers and computer users was as networked as it is now, there were *multiple independent and parallel hacker subcultures*, often unaware or only partially aware of each others' existence. All of these had *certain important traits* in common:

- creating software and sharing it with each other
- placing a high value on freedom of inquiry; hostility to secrecy
- information-sharing as both an ideal and a practical strategy
- upholding the right to fork
- emphasis on rationality
- distaste for authority
- playful cleverness, taking the serious humorously and their humor seriously

2. These sorts of subcultures were commonly found at *academic settings* such as college campuses. *The MIT Artificial Intelligence Laboratory, the University of California, Berkeley and Carnegie Mellon University* were particularly *well-known hotbeds of early hacker culture*. Over time, the academic hacker subculture has tended to become more *conscious, more cohesive, and better organized.* The most important consciousness-raising moments have included *the composition of the first Jargon File* in 1973, *the promulgation of the*

GNU Manifesto in 1985, and the publication of *The Cathedral and the Bazaar* in 1997. Correlated with this has been the gradual election of a set of shared culture heroes: *Bill Joy, Donald Knuth, Dennis Ritchie, Alan Kay, Ken Thompson, Richard M. Stallman, Linus Torvalds, and Larry Wall,* among others.

3. The concentration of academic hacker subculture has paralleled and partly been driven by *the commoditization of computer and networking technology*, and has in turn accelerated that process. In 1975, hackerdom was scattered across several different families of operating systems and disparate networks; today it is largely *a Unix and TCP/IP phenomenon*, and is concentrated around *various open-source operating systems*. *The academic hacker subculture* is defined by *shared work and play focused around central artifacts*. Some of these artifacts are very large; the Internet itself, the World Wide Web, the GNU Project, and the Linux operating system are all hacker creations, works of which the subculture considers itself primary custodian. Since 1990, *the academic hacker subculture* has developed a rich range of *symbols* that serve as recognition symbols and reinforce its group identity. *Tux, the Linux penguin, the BSD Daemon, and the Perl Camel* stand out as examples. All of these routinely adorn T-shirts, mugs, and other paraphernalia. Notably, the academic hacker subculture appears to have exactly one annual ceremonial day—*April Fool's.* There is a long tradition of perpetrating elaborate jokes, hoaxes, pranks and fake websites on this date. This is so well established that hackers look forward every year to the publication of the *annual joke RFC*, and one is invariably produced.

inquiry – вопрос, запрос

uphold – помогать, поддерживать

to fork – *зд.* отделяться

distaste – отвращение

hotbed – очаг

***cohesive* –** сплоченный

promulgation – опубликование, пропаганда

reinforce – укреплять, повторить с новыми силами

***adorn* –** украшать

paraphernalia - атрибуты

TEXT 24

ASSIGNMENT 1.

Look through the following text. Find its main information concentrating on the beginning of each paragraph and following the key words with the words related to them.

ASSIGNMENT 2.

Look through paragraph 1.

Concentrate on the proper names and dates so as to answer the question: *Where did Babbage study to get his education?*

ASSIGNMENT 3.

Read paragraph 2. Concentrate on the details so as to answer the questions:

1. What factors influenced Babbage in his work?

2. What devices did Babbage develop?

ASSIGNMENT 4.

Read paragraph 3. Enumerate all main points concentrating on the proper names and dates. Repetition of the same word may be helpful.

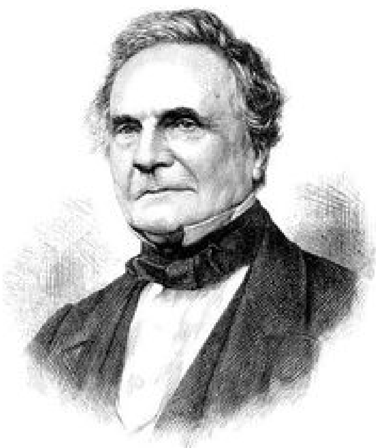
Answer the questions: *Why wasn't Babbage's discovery accepted?*

ASSIGNMENT 5.

Give a summary of the text.

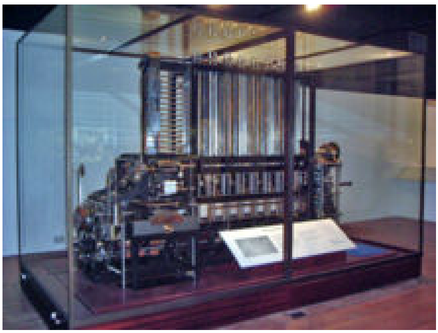
Charles Babbage

4259 t. un.



1. *Charles Babbage* (26 December 1791 – 18 October 1871) was *an English mathematician, philosopher, mechanical engineer and (proto-) computer scientist* who originated the idea of a *programmable* computer. Charles Babbage was born in England. Charles's father, Benjamin Babbage, was a banking partner of the Praeds who owned the Bitton Estate in

Teignmouth. His mother was Betsy Plumleigh Babbage née Teape. His father's money allowed Charles to receive instruction from several schools and tutors during the course of his elementary education. For a short time he attended *King Edward VI Grammar School* in Totnes, South Devon, but his health forced him back to private tutors for a time. He then joined a 30-student *Holmwood academy*, in Baker Street, Enfield, Middlesex under Reverend Stephen Freeman. The academy had a well-stocked library that prompted Babbage's love of mathematics. He studied with two more private tutors after leaving the academy. Babbage arrived at *Trinity College, Cambridge* in October 1810. He had read extensively in Leibniz, Lagrange, Simpson, and Lacroix and was seriously disappointed in the mathematical instruction available at Cambridge. In response, he, *John Herschel, George Peacock,* and several other friends formed *the Analytical Society* in 1812. In 1812 Babbage transferred to *Peterhouse, Cambridge*. He was the top mathematician at Peterhouse, but failed to graduate with honours. He instead received an honorary degree without examination in 1814.



2. In recognition of the high error rate in the calculation of mathematical tables, Babbage wanted to find *a method* by which they could be calculated *mechanically*, removing human sources of error. *Three different factors* seem to have influenced him: a dislike of untidiness; his experience working on logarithmic tables; and existing work on calculating machines carried out by Wilhelm Schickard, Blaise Pascal,

and Gottfried Leibniz. He first discussed the principles of *a calculating engine* in a letter to Sir Humphry Davy in 1822. Babbage's engines were among *the first mechanical computers*. His *engines* were not actually completed, largely because of *funding problems and personality issues.* Babbage realized that a machine could do the work better and more reliably than a human being. Babbage controlled building of some *steam-powered machines* that more or less did their job; calculations could be mechanized to an extent. Although Babbage's machines were mechanical monsters their *basic architecture* was astonishingly similar to *a modern computer*. The data and program memory were separated, operation was instruction based, control unit could make conditional jumps and the machine had a separate I/O unit. Babbage designed *a printer* for the second difference engine which had some remarkable features; it supported line-wrapping, variable column and row width, and programmable output formatting.

3. In 1824, *Babbage* won *the Gold Medal of the Royal Astronomical Society* "for his invention of an engine for calculating mathematical and astronomical tables". From 1828 to 1839 Babbage was *Lucasian professor of mathematics at Cambridge*. He contributed largely to several scientific periodicals, and was instrumental in founding *the Astronomical Society* in 1820 and *the Statistical Society* in 1834. However, he dreamt of designing mechanical calculating machines. Charles Babbage also achieved notable results in *cryptography*. He broke Vigenère's autokey cipher as well as the much weaker cipher that is called Vigenère cipher today. Babbage's discovery was used to aid *English military campaigns*, and was not published until several years later; as a result credit for the

development was instead given to *Friedrich Kasiski,* a Prussian infantry officer, who made the same discovery some years after Babbage. Babbage also invented *the pilot* (also called *a cow-catcher*), the metal frame attached to the front of locomotives that clears the tracks of obstacles in 1838. He also constructed *a dynamometer car* and performed several studies on Isambard Kingdom Brunel's Great Western Railway in about 1838. Babbage is also credited with the invention of *standard railroad gauge*, *uniform postal rates*, *occulting lights for lighthouses, the heliograph, and the ophthalmoscope.*

🔔 ***née*** – урожденная

reverend – почтенный

gauge – измерительный прибор

TEXT 25

ASSIGNMENT 1.

Look through the following text. Follow the key words and the words related to them. Concentrate on the signals. Give the main information of the text.

ASSIGNMENT 2.

Look through the text more carefully.

Answer the questions:

In paragraph 1: *Were did Hollerith study?*

In paragraph 2: *What were the basic principals of his tabulating machine?*

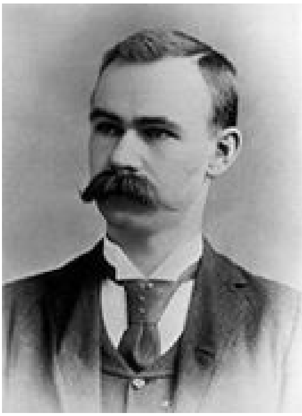
In paragraph 3: *How did his career develop after his invention?*

ASSIGNMENT 3.

Give a summary of the text.

Herman Hollerith

3846 t. un.



1. *Herman Hollerith* (February 29, 1860 – November 17, 1929) was *an American statistician* who developed *a mechanical tabulator based on punched cards* to rapidly tabulate statistics from millions of pieces of data. He was born on February 29, 1860, *in Buffalo, New York*, to Johann Georg Hollerith (1808–1869) and Franciska Brunn, both of Rhineland-Palatinate, Germany. He entered *the City College of New York* in 1875 and graduated from the *Columbia University School of Mines* with an "Engineer of Mines" degree in 1879. In 1880, he listed himself as a mining engineer while living in Manhattan, and he completed his Ph.D. in 1890 at *Columbia University*. In 1890, on September 15, he married Lucia Beverley Talcott (December 3, 1865 – August 4, 1944) of Vera Cruz, Mexico, and they had six children (three sons and three daughters). Other than his inventions, Hollerith was said to cherish three things: his German

heritage, his privacy and his cat, Bismarck. He died in 1929 of a heart attack and was buried in *the Oak Hill Cemetery in Georgetown, Washington, D.C.*

2. After graduating, he immediately obtained a job with *the US Census Bureau* as *a special agent* collecting and analyzing statistical information on the use of steam and water power in the iron and steel industries. This was the inspiration for his one obsession as an inventor, and indeed for his one successful invention; his other ones were notably unsuccessful. Although he subsequently moved to *Massachusetts Institute of Technology* as *an instructor*, he still continued to research *a device for recording census statistics*. He knew of *the Jacquard loom* which used holes in cards to program its complicated patterns of weaving, but it was reportedly the further inspiration of *a 'punch photograph' train ticket*, on which passenger details (such as height and hair color) were punched out around the edge by the conductor, that clinched his *key invention*. Hollerith decided that each census taker could do the same, with the resulting punched card being sorted by a variation of the Jacquard loom; the important aspects of the invention being that the holes were sensed electrically rather than mechanically and that one card held all the information on an individual.

3. Thus, urged on by *John Shaw Billings*, he developed *a mechanism for reading the presence or absence of holes in the cards* using spring-mounted needles that passed through the holes to make electrical connections to trigger a counter to record one more of each value. *The key idea* (due to Billings), however, was that *all personal data could be coded numerically*. Hollerith saw that if the numbers could then be punched in specified columns on the cards, the cards

could be sorted mechanically, and therefore the appropriate columns totaled. He built machines under contract for *the US Census Bureau*, which used them to tabulate the 1890 census in 2.5 years. The 1880 census had taken 7 years to complete. He started his own business in 1896 when he founded the *Tabulating Machine Company*. Most of the major census bureaus around the world leased his equipment and purchased his cards, as did major insurance companies. To make his system work he invented *the first automatic card-feed mechanism*, *the first Key punch* (i.e. a punch that was operated from a keyboard) allowing a skilled operator to punch 200–300 cards per hour and a tabulator. The 1890 Tabulator was hardwired to operate only on 1890 Census cards. A wiring panel in his 1906 *Type I Tabulator* allowing it to do different jobs without having to be rebuilt (the first step towards programming).These inventions were the foundation of *the modern information processing industry*. In 1911, his firm merged with two others to form *the Computing Tabulating Recording Corporation (CTR)*. Under the presidency of Thomas J. Watson, it was renamed *IBM* in 1924.

heritage - наследие
obsession – одержимость, навязчивая идея
edge – край, кромка
to clinch – *зд.* окончательно решить, урегулировать
to urge – побуждать, убеждать
spring-mounted needles – *зд.* встроенные подъемные иглы
to trigger – привести в действие, инициировать

Костина Елена Витальевна

READING FOR INFORMATION AND LEARNING

Обучение ознакомительному чтению

Методическое пособие

Компьютерный набор: Е.В. Костина
Технический редактор: Г.В. Куликова

ФГБОУ ВО Ивановский государственный
химико-технологический университет

153000, г Иваново, Шереметевский пр., 7