**Задание 1.** Операционная система. Функции операционных систем. Понятие файла, файловой структуры. Операционная система Windows: организация хранения файлов на дисках.

Операционная система — это комплекс взаимосвязанных системных программ, назначение которого — организовать взаимодействие пользователя с компьютером и выполнение всех других программ  
Операционная система выполняет роль связующего звена между аппаратурой компьютера, с одной стороны, и выполняемыми программами, а также пользователем, с другой стороны.

Операционная система обычно хранится во внешней памяти компьютера — на диске. При включении компьютера она считывается с дисковой памяти и размещается в ОЗУ. Этот процесс называется загрузкой операционной системы.

В функции операционной системы входит:

* осуществление диалога с пользователем;
* ввод-вывод и управление данными;
* планирование и организация процесса обработки программ;   
  распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
* запуск программ на выполнение;
* всевозможные вспомогательные операции обслуживания;
* передача информации между различными внутренними устройствами;
* программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

Операционную систему можно назвать программным продолжением устройства управления компьютера. Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера.  
В зависимости от количества одновременно обрабатываемых задач и числа пользователей, которых могут обслуживать ОС, различают четыре основных класса операционных систем:

* однопользовательские однозадачные, которые поддерживают одну клавиатуру и могут работать только с одной (в данный момент) задачей;
* однопользовательские однозадачные с фоновой печатью, которые позволяют помимо основной задачи запускать одну дополнительную задачу, ориентированную, как правило, на вывод информации на печать. Это ускоряет работу при выдаче больших объёмов информации на печать;
* однопользовательские многозадачные, которые обеспечивают одному пользователю параллельную обработку нескольких задач. Например, к одному компьютеру можно подключить несколько принтеров, каждый из которых будет работать на "свою" задачу;
* многопользовательские многозадачные, позволяющие на одном компьютере запускать несколько задач нескольким пользователям. Эти ОС очень сложны и требуют значительных машинных ресурсов.

В различных моделях компьютеров используют операционные системы с разной архитектурой и возможностями. Для их работы требуются разные ресурсы. Они предоставляют разную степень сервиса для программирования и работы с готовыми программами.

Операционная система для персонального компьютера, ориентированного на профессиональное применение, должна содержать следующие основные компоненты:

* программы управления вводом/выводом;
* программы, управляющие файловой системой и планирующие задания для компьютера;
* процессор командного языка, который принимает, анализирует и выполняет команды, адресованные операционной системе.

Каждая операционная система имеет свой командный язык, который позволяет пользователю выполнять те или иные действия:

* обращаться к каталогу;
* выполнять разметку внешних носителей;
* запускать программы;

... другие действия.

Анализ и исполнение команд пользователя, включая загрузку готовых программ из файлов в оперативную память и их запуск, осуществляет командный процессор операционной системы.

Для управления внешними устройствами компьютера используются специальные системные программы — драйверы. Драйверы стандартных устройств образуют в совокупности базовую систему ввода-вывода (BIOS), которая обычно заносится в постоянное ЗУ компьютера.

Файл (англ. file —папка) — это именованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и др.   
Файловая система — это средство для организации хранения файлов на каком-либо носителе.

Файлы физически реализуются как участки памяти на внешних носителях — магнитных дисках или CD-ROM. Каждый файл занимает некоторое количество блоков дисковой памяти. Обычная длина блока — 512 байт.   
Обслуживает файлы специальный модуль операционной системы, называемый драйвером файловой системы. Каждый файл имеет имя, зарегистрированное в каталоге — оглавлении файлов.

Каталог (иногда называется директорией или папкой) доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нем файлы, переносить их содержимое на новое место и удалять.

Каталог может иметь собственное имя и храниться в другом каталоге наряду с обычными файлами: так образуются иерархические файловые структуры.   
Драйвер файловой системы обеспечивает доступ к информации, записанной на магнитный диск, по имени файла и распределяет пространство на магнитном диске между файламиК файловой системе имеет доступ также и любая прикладная программа, для чего во всех языках программирования имеются специальные процедуры.

Понятие файла может быть обращено на любой источник или потребитель информации в машине, например, в качестве файла для программы могут выступать принтер, дисплей, клавиатура и др.

Структура файловой системы и структура хранения данных на внешних магнитных носителях определяет удобство работы пользователя, скорость доступа к файлам и т.д.

Навигация по файловой структуре является одной из наиболее используемых функций операционной системы. Удобство этой операции часто воспринимают как удобство работы с операционной системой. В операционных системах, имеющих интерфейс командной строки, навигацию осуществляют путем ввода команд перехода с диска на диск или из каталога в каталог. В связи с крайним неудобством такой навигации, широкое применение нашли специальные служебные программы, называемые файловыми оболочками.  
Как и операционные системы, файловые оболочки бывают неграфическими и графическими. Наиболее известная неграфическая файловая оболочка для MS-DOS — диспетчер файлов Norton Commander, а роль графической файловой оболочки для MS-DOS в свое время исполняли программы Windows 1.0 и Windows 2.0, которые постепенно развились до понятия операционной среды (в версиях Windows 3.x) и далее до самостоятельной операционной системы (Windows 95/98).

### Загрузочная запись

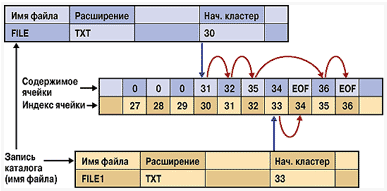
Одна из наиболее значимых информационных структур логического диска — его загрузочная запись. Она располагается в первом секторе диска (с появлением Windows 95 OSR2 при использовании FAT32 загрузочная запись размещается в нескольких секторах).

Назначение данной записи — загрузка ОС и организация хранения данных. Если логический диск является системным, то в его первом секторе имеется код загрузки ОС, управляющийся от MBR (Master boot record). Его задача — загрузка основных файлов ОС и передача им управления; в Windows 9х такими файлами будут io.sys и msdos.sys.

Если не существует загрузочной записи, то она создается и заполняется программой FORMAT. Но если она уже есть, то утилита FORMAT будет использовать ее информацию.

### Таблица размещения файлов

Следующая ступень организации хранения данных на жестком диске — FAT (File Allocation Table — таблица размещения файлов). Она, как правило, представлена в двух экземплярах, следующих друг за другом и содержащих одинаковую информацию (при условии, что все в порядке). Нужно помнить, что операционные системы DOS и Windows не различают цилиндры, головки и физические секторы диска, который для этих ОС предстает в виде непрерывной последовательности логических секторов или кластеров (группы смежных секторов). Система MS-DOS до версии 7.0 включительно и система Windows могли распознать 65 536 логических блоков на диске. Начиная с Windows 95 OSR2, появилась возможность использовать для нумерации логических элементов на диске 32-разрядные данные, а значит, число адресуемых элементов теоретически возросло до 4 294 967 295. Для DOS до версии 3.х включительно размер логического блока равнялся 512 байтам, т. е. размеру физического сектора. Видимо, поэтому появилось понятие «логический сектор», используемое и поныне. Наибольшая емкость диска, с которым могла работать DOS, составляла 32 Мбайт. Но был найден выход: объединять логические секторы в группы — так называемые кластеры. Под этим термином понимается группа таких смежных секторов, которым соответствует одно значение адреса. Если увеличить размер кластера, то можно будет работать с большими разделами, оставаясь в рамках 16-разрядной адресации. После перехода к 32-разрядной адресации стало возможным (правда, лишь теоретически) применять кластеры любого размера на любых разделах.



Идея FAT очень проста. Все пространство логического диска разбивается на кластеры, размер которых зависит от емкости диска. Затем составляется таблица, каждому элементу которой соответствует элемент дискового пространства — кластер. Эта таблица линейная: индекс ячейки соответствует номеру кластера.

Если исключить резервирование кластеров, то максимальное число адресуемых кластеров в FAT16 — 65 526, в FAT32 — 4 294 377 472. Последние четыре разряда каждой записи FAT32 пока зарезервированы.

### Корневой каталог

Логически корневой каталог служит оглавлением — это следующая за второй FAT область на логическом диске, являющаяся последовательностью 32-байтовых записей. Каждая из последних может быть каталогом, именем файла (в том числе и длинным), а также меткой тома. Нужно сразу отметить, что, в отличие от FAT32, в FAT16 размер корневого каталога зафиксирован.

В FAT16 таких записей 512. Легко подсчитать: 512х32 = 16 384 байта, или 32 сектора. При использовании FAT32 корневой каталог рассматривается как обычный каталог и его размер растет по мере необходимости.

Под файлом данных понимается поименованная последовательность байтов. Причем ИМЯ хранится отдельно от этой последовательности. Операционная система работает с данными через ИМЯ, которое есть 32-байтовая запись. В ней закодированы текстовое имя, размер, дата, время, атрибуты и номер первого кластера.

Несколько слов об атрибутах файлов. Вот основные из них: «только чтение», «скрытый», «системный», «метка тома», «подкаталог», «архивный». Атрибут «подкаталог» сообщает ОС, что данная запись относится к подкаталогу, атрибут «метка тома» — к метке тома. Остальные атрибуты относятся к файлам данных и указывают, как система должна с ними работать.

**Задание 2.**

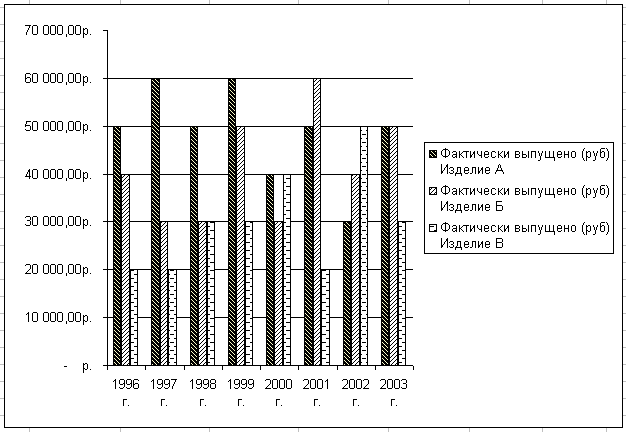
Формулировка задания:

1. Выполнить задание, общее для всех вариантов.
2. В ячейке F14 рассчитать наименьшее значение фактического выпуска Изделия Б.
3. В ячейке F15 рассчитать количество лет, за которые фактический выпуск Изделия А составил меньше 50000 руб.
4. Построить гистограмму, характеризующую соотношение по годам значений фактического выпуска Изделия А, Изделия Б, Изделия В (на одной диаграмме три ряда данных).

Расчетная таблица:



Диаграмма:



Описание формул, используемых в расчетах:

1. **СУММ()**: суммирует все числа в интервале ячеек.

Синтаксис

СУММ(число1; число2; ...)

Число1, число2, ... — от 1 до 30 аргументов, для которых требуется определить итог или сумму.

1. **Если()**: возвращает одно значение, если заданное условие при вычислении дает значение ИСТИНА, и другое значение, если ЛОЖЬ.

Функция ЕСЛИ используется при проверке условий для значений и формул.

Синтаксис

ЕСЛИ(лог\_выражение; значение\_если\_истина; значение\_если\_ложь)

Лог\_выражение  — это любое значение или выражение, принимающее значения ИСТИНА или ЛОЖЬ. Например, F3>=B3 — это логическое выражение; если значение в ячейке F3 больше или равно значению в ячейке B3, то выражение принимает значение ИСТИНА. В противном случае — ЛОЖЬ.

Значение\_если\_истина   — это значение, которое возвращается, если лог\_выражение равно ИСТИНА. Например, если этот аргумент — строка «выполнен» и лог\_выражение равно ИСТИНА, тогда функция ЕСЛИ отобразит текст «выполнен».

Значение\_если\_ложь   — это значение, которое возвращается, если лог\_выражение равно ЛОЖЬ. Например, если этот аргумент — строка «не выполнен» и лог\_выражение равно ЛОЖЬ, то функция ЕСЛИ отобразит текст «не выполнен»..

1. **СРЗНАЧ():** возвращает среднее (арифметическое) своих аргументов.

Синтаксис

СРЗНАЧ(число1; число2; ...)

Число1, число2, ...   — это от 1 до 30 аргументов, для которых вычисляется среднее.

1. **МИН():** возвращает наименьшее значение в списке аргументов.

Синтаксис

МИН(число1;число2; ...)

Число1, число2, ... — от 1 до 30 чисел, среди которых требуется найти наименьшее.

1. СЧЕТЕСЛИ(): подсчитывает количество ячеек внутри диапазона, удовлетворяющих заданному критерию.

Синтаксис

СЧЁТЕСЛИ(диапазон;критерий)

Диапазон — диапазон, в котором нужно подсчитать ячейки.

Критерий  — критерий в форме числа, выражения или текста, который определяет, какие ячейки надо подсчитывать. Например, критерий может быть выражен следующим образом: "<50000".

**Задание 3.**

Задача 1. Составить программу расчета значений функции  на интервале изменения , равном  в  равноотстоящих точках. Вычисленные значения сохранить в массиве . А также найти сумму положительных значений функции  в расчетных точках. 

Решение.

Таблица обозначений в программе:

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Обозначение** | **Тип** |
| Аргумент функции | **x** | **Real** |
| Границы интервала изменения x | **a, b** | **Real** |
| Количество точек | **n** | **Integer** |
| Шаг изменения аргумента x | **dx** | **Real** |
| Текущий номер расчетной точки | **i** | **Integer** |
| Массив, содержащий вычисленные значения функции | **c** | **Real** |
| Сумма положительных значений функции | **sum** | **Real** |
| Вспомогательная переменная | **c** | **Char** |

Текст программы:

Program task1;

Var x, a, b, dx, sum: Real;

n, i: Integer;

c: Array[1..100] of Real;

symbol: Char;

Begin

Write('number of points: '); readln(n);

Write('a, b (a<b): '); readln(a, b);

dx := (b-a)/(n-1); sum := 0;

For i:=1 To n Do Begin

x := a + (i-1)\*dx;

c[i] := 5\*exp(0.5\*x)\*sin(Pi\*x);

Writeln(x:8:3, c[i]:15:3);

If c[i] > 0 Then sum := sum + c[i]

End;

Writeln('sum = ', sum:8:3);

readln(symbol)

End.

Задача 2. Для произвольного двумерного массива , содержащего значения типа Real, состоящего из  строк и  столбцов, сформировать одномерный массив , каждый элемент которого равен произведению значений элементов соответствующей строки массива , целая часть которых делится на 3 без остатка.

Решение.

Таблица обозначений в программе:

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Обозначение** | **Тип** |
| Массивы | **a, b** | **Real** |
| Количество строк (столбцов) массива a | **n** | **Integer** |
| Индексы | **i, j** | **Integer** |
| Указатель наличия в соответствующей строке массива  элементов, целая часть которых делится на 3 без остатка | **key** | **Boolean** |
| Вспомогательная переменная | **c** | **Char** |

Текст программы:

Program task2;

Var a: Array[1..10, 1..10] of Real;

b: Array[1..10] of Real;

n, i, j: Integer;

key: Boolean;

symbol: Char;

Begin

Write('rows number of array a: '); readln(n);

Writeln(''); Write(' ');

For j:=1 To n Do write(j:7);

Writeln('');

For i:=1 To n Do Begin

Write('Row ', i:2, ' ');

For j:=1To n Do Read(a[i, j])

End;

Writeln(''); Write('b: ');

For i:=1 To n Do Begin

b[i] := 1; key := False;

For j:=1 To n Do

If Trunc(a[i, j]) mod 3= 0 Then Begin

b[i] := b[i] \* a[i, j]; key := True

End;

If Not key Then b[i] := 0;

Write(b[i]:7:2);

End;

readln(symbol)

End.