

## ЛАБОРАТОРНАЯ РАБОТА № 1

### Встроенные типы данных.

Выполнить с двумя вещественными аргументами, принадлежащими отрезку [0..9]:

- 1) Все арифметические операции
- 2) Преобразования в целые значения с помощью функций TRUNC и ROUND и все операции целочисленной арифметики
- 3) преобразование целых значений в символьный с использованием функций CHR и ORD

В случае непринадлежности переменных вышеуказанному промежутку запрос на ввод данных переменных повторяется до тех пор, пока каждое из них не будет удовлетворять условию принадлежности этому промежутку.

В случае деления на 0 операция не производится.

Исходные данные:

- |          |       |
|----------|-------|
| 1) a=0   | b=5.1 |
| 2) a=7.2 | b=2.4 |
| 3) a=4.7 | b=0   |

## ЛАБОРАТОРНАЯ РАБОТА № 2

### Программирование операций ввода-вывода.

В созданном вами текстовом файле удалить запись с заданным номером.

Исходные данные:

f : text – файл, над которым требуется совершить вышеуказанную операцию (test.txt);

n : integer – номер строки, которую необходимо удалить.

Выходные данные:

f : text – файл после совершения над ним операции удаления записи с заданным номером (test.txt).

## ЛАБОРАТОРНАЯ РАБОТА № 3

### Целочисленная арифметика.

Выполнить перевод числа из 7-ичной системы счисления в 2-ичную и из 2-ичной – в 3-ичную систему счисления. Число должно быть неотрицательным.

Исходные данные:

InputSTR : string – строка, содержащая изображение числа в 7-ичной системе счисления.

Выходные данные:

Result1 : longint – то же число в 2-ичной системе счисления;

Result2 : string – строка, содержащая то же число в 3-ичной системе счисления.

### Генератор псевдослучайных чисел

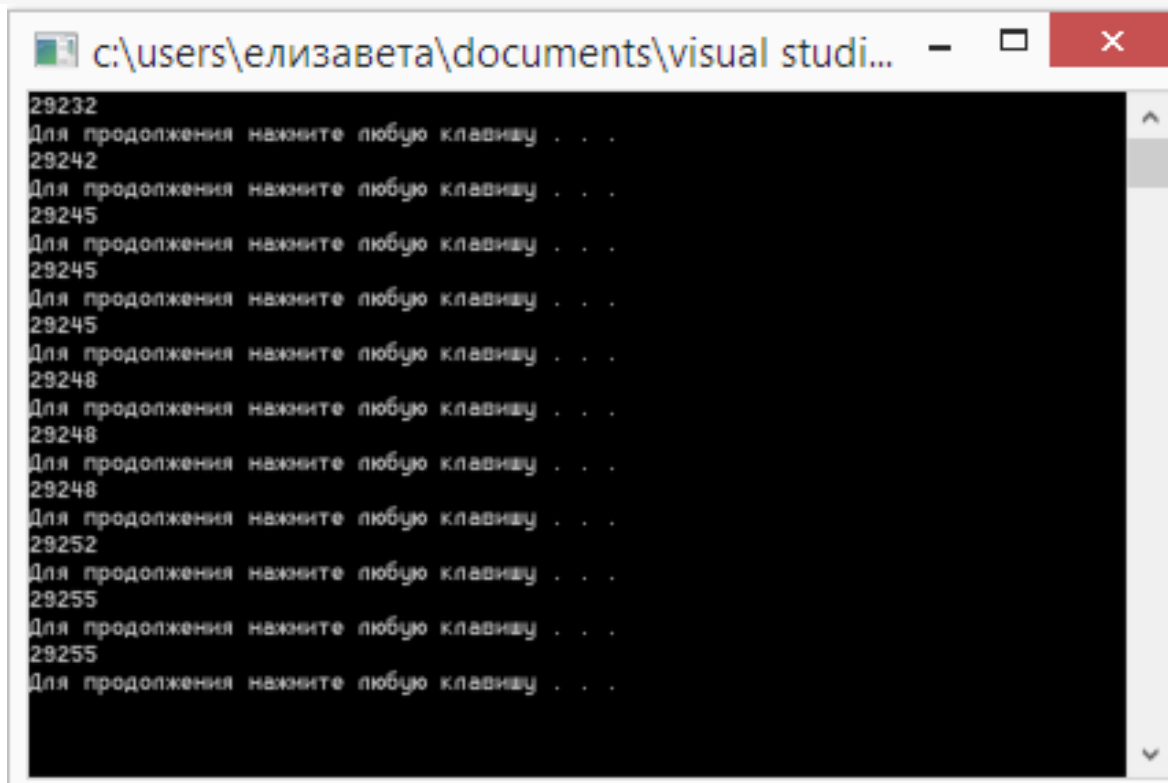
```
#include <cstdlib>
#include <ctime>
#include <iostream>
using namespace std;

void main() {
    while(1) {
```

```

srand(time(NULL));
cout << rand() << endl;
system("pause");
}
}

```



```

// Инициализация генератора случайных чисел
srand(time(0));
//Генерация случайного числа диапазон от 0 до 10;
int r;
r = rand() % 10;

```

## ЛАБОРАТОРНАЯ РАБОТА № 4

### Генератор случайных вещественных чисел

```

#include "stdafx.h"
#include <cstdlib>
#include <ctime>
#include <cmath>
#include <cstring>
#include <iostream>
#include <iomanip>

int main()
{
    srand(time(0));

    for (int i = 0; i < 5; ++i) {
        double randomValue = (rand() % 100000) / 10000.0; // число с пятью знаками после
запятой
        for (int j = 5; j > 0; --j) {
            std::cout << "PRECISION IS " << j << ": " <<

```

```

        std::fixed << std::setprecision(j) << randomValue << std::endl;
    }
}

std::cin.peek();
return 0;
}

```

```

PRECISION IS 4: 3.2347
PRECISION IS 3: 3.235
PRECISION IS 2: 3.23
PRECISION IS 1: 3.2
PRECISION IS 5: 2.23520
PRECISION IS 4: 2.2352
PRECISION IS 3: 2.235
PRECISION IS 2: 2.24
PRECISION IS 1: 2.2
PRECISION IS 5: 2.00030
PRECISION IS 4: 2.0003
PRECISION IS 3: 2.000
PRECISION IS 2: 2.00
PRECISION IS 1: 2.0
PRECISION IS 5: 2.40740
PRECISION IS 4: 2.4074
PRECISION IS 3: 2.407
PRECISION IS 2: 2.41
PRECISION IS 1: 2.4
PRECISION IS 5: 0.94420
PRECISION IS 4: 0.9442
PRECISION IS 3: 0.944
PRECISION IS 2: 0.94
PRECISION IS 1: 0.9

```

## Вещественная арифметика.

Вычислить число  $\pi$  двумя методами – статистическим и геометрическим. Для статистического метода число вбрасываемых в квадрат точек должно удовлетворять интервалу  $[1..1000000]$  (большее число раз цикл будет выполняться очень долго). Для геометрического метода число удвоений сторон должно удовлетворять интервалу  $[1..31]$  (при большем значении конечное количество сторон превысит вместимость `longint`).

Исходные данные:

Count1 : `longint` – число вбрасываемых в квадрат точек для статистического метода;

Count2 : `longint` – число удвоений кол-ва сторон для геометрического метода.

Выходные данные:

Pi1 : `real` – число  $\pi$ , вычисленное с помощью статистического метода;

Pi2 : `real` – число  $\pi$ , вычисленное с помощью статистического метода.

Исходные данные:

- 1) Count1=16; Count2=16.
- 2) Count1=1000000; Count2=31.
- 3) Count1=10000000; Count2=100.

Выходные данные:

- 1) Pi1=(от 2,25 до 3,75); Pi2=3,141592652.
- 2) Pi1=3,1412; Pi2=3,1415926536.

Описание алгоритма

### 1) Вычисление числа $\pi$ при помощи статистического метода.

Полагая  $\pi$  равным отношению  $4 \cdot S_{\text{кр}} / S_{\text{кв}}$ , где  $S_{\text{кв}}$  – площадь четвертой части квадрата со стороной, равной 2, а  $S_{\text{кр}}$  – площадь сектора, составляющего четвертую часть круга, вписанного в вышеуказанный квадрат, оцениваем  $S_{\text{кр}}$  и  $S_{\text{кв}}$  количеством случайных точек, попавших в сектор и в часть квадрата соответственно. Положение каждой случайной точки определяем с помощью генератора псевдослучайных чисел, интерпретируя пару случайных чисел как координаты точки в двумерной системе координат с началом в центре круга.

### 2) Вычисление числа $\pi$ при помощи геометрического метода.

Полагая  $\pi$  равным половине длины окружности единичного радиуса, оцениваем его произведением длины стороны ломаной, вписанной в эту дугу, на количество сторон этой ломаной. Каждое очередное значение длины стороны ломаной получаем из предыдущего, воспользовавшись при этом рекуррентным соотношением:

$$x = \sqrt{\left(\frac{x}{2}\right)^2 + \left(1 - \sqrt{1 - \left(\frac{x}{2}\right)^2}\right)^2},$$

где  $x$  – длина стороны ломаной.

## ЛАБОРАТОРНАЯ РАБОТА № 5

### Действия над множествами

Построить множество всех простых чисел, меньших заданного числа  $n$ , с помощью алгоритма, известного под названием "Решето Эратосфена". В качестве исходного, предназначенного для просеивания, множества выбрать структуру типа `set`.

#### Описание алгоритма

Основной идеей алгоритма "Решето Эратосфена" является удаление из исходного множества всех имеющихся в нём составных чисел: сначала - кратных 2, затем - кратных 3 и т.д. На  $i$ -том шаге алгоритма из множества удаляются числа, кратные числу  $j$ , не удаленному ранее и следующему за простым  $k$ , кратность которому была признаком удаления чисел на  $(i-1)$ -ом шаге алгоритма.

## ЛАБОРАТОРНАЯ РАБОТА № 6

### Операции над списковыми структурами

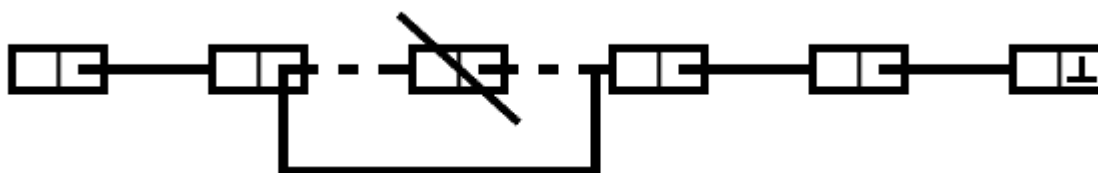
Написать программу построения линейной списковой структуры, содержащей данные типа `string`, а также процедуру, реализующую удаление из списка записи с заданным значением поля данных. Линейная списковая структура формируется из последовательности записей текстового файла.

Исходные данные:

- указатель на первый элемент неизменённого списка;
- строка со значением поля данных записи, которую необходимо удалить.

### Описание алгоритма

Построение линейной списковой структуры выполняется путём последовательного считывания строк из текстового файла и занесения их в поля данных очередных записей списка. Удаление из списка записи с заданным значением поля данных реализуется с помощью переноса значения поля указателя из удаляемой записи в предыдущую в случае, если подлежащая удалению запись не первая в списке. В противном случае 2-ой элемент считается первым, и список перезадаётся указателем на него. В любом случае память, выделенная под хранение удаляемой



записи, освобождается.

## ЛАБОРАТОРНАЯ РАБОТА № 7

### Строки и массивы

Написать в среде Visual C++ консольную программу сортировки массива строк с помощью алгоритма прямого выбора. Ввод данных организовать с клавиатуры, вывод результата – на экран.

#### 1. Описание входных и выходных данных

Исходные данные:

`char *S_Arr[]` – массив указателей на неупорядоченные строки;  
`int S_Amt` – количество строк.

Выходные данные:

`char *S_Arr[]` – массив указателей на упорядоченные строки.

#### 2. Набор тестов

Исходные данные:

`S_Arr[1]` указывает на "ACBD"

`S_Arr[2]` указывает на "DCBA"

*S\_Arr[3] указывает на "ABCD"*

*S\_Arr[4] указывает на "BDAC"*

*S\_Arr[5] указывает на "CDBA"*

Выходные данные:

*S\_Arr[1] указывает на "ABCD"*

*S\_Arr[2] указывает на "ACBD"*

*S\_Arr[3] указывает на "BDAC"*

*S\_Arr[4] указывает на "CDBA"*

*S\_Arr[5] указывает на "DCBA"*

### 3. Описание алгоритма

Основная идея алгоритма прямого выбора – на *i*-ом шаге среди текущей последовательности (от (*i*+1)-ого элемента до *n*-ого элемента) отыскивается минимальный элемент и меняется с *i*-ым элементом.

### 4. Текст программы

```
#include <iostream.h>
#include <malloc.h>
#include <string.h>

//const
const int S_Amt = 5; //count of strings in array
//var
char *S_Arr[S_Amt+1]; //array which needs sorting

void initialization (char *S_Arr[], int n)
{
//var
int i; //parameter of cycle

//program does not use 0-element of array
S_Arr[0] = (char *) malloc(256);
S_Arr[0] = "Zero";
//taking memory for i-element
for (i=1; i<=S_Amt; i++) S_Arr[i] = (char *) malloc(256);
}

void input (char *S_Arr[], int S_Amt)
{
int i; //parameter of cycle
//array is being taken in...
for (i=1; i<=S_Amt; i++)
{
cout<<"Input "<<i<<" element of array -> ";
//inputing i-element
cin>>S_Arr[i];
}
}
```

```

void MoSS(char *S_Arr[], int S_Amt)
{
//var
    char *x; //buffer string for exchange
    int i;    //parameter of first cycle
    int j;    //parameter of second cycle
    int k;    //number of least string in array

    //initialization
    x = (char *) malloc (256);
    //first cycle
    for(i=1; i<S_Amt; i++)
    {
        //i-string is current
        k=i;
        //the current string is being copied into buffer...
        strcpy(x,S_Arr[i]);
        //search of least string in array by means second cycle
        for (j=i+1; j<=S_Amt; j++)
        {
            if (strcmp(S_Arr[j],x)<0)
            {
                k=j;
                //string which is less than the string in buffer is becoming current...
                strcpy(x,S_Arr[k]);
            }
        }
        //if the search has been successful then the string in buffer and the current
        string are being swapped...
        if (k!=i)
        {
            strcpy(S_Arr[k],S_Arr[i]);
            strcpy(S_Arr[i],x);
        }
    }
}

void output (char *S_Arr[], int S_Amt)
{
//var
    int i; //parameter of cycle

    cout<<"The modified array is:"<<"\n";
    //outputting i-element
    for (i=1; i<=S_Amt; i++) cout<<S_Arr[i]<<"\n";
}

void main()
{
    //preparing array to work...
    initialization(S_Arr,S_Amt);
    //the array is being taken in...
    input(S_Arr,S_Amt);
    //sort2 is sorting the array by means the method of straight selecting...
    MoSS(S_Arr,S_Amt);
    //the array is being put out...
    output(S_Arr,S_Amt);
}

```

## 5. Анализ результатов и выводы

Программа корректно работает и выполняет поставленную задачу. Объём программы составил 188472 байт. Для  $n$  элементов:

количество сравнений

$$P = (n-1) + (n-1) + (n-2) + \dots + 1 \quad (\text{число порядка } n^2),$$

среднее количество перестановок элементов

$$N_{cp} = n * (\ln(n) + 0,577) .$$

## ЛАБОРАТОРНАЯ РАБОТА № 8

### Работа с записями

#### 1. Постановка задачи

В среде Visual C++ написать консольную программу для работы с табличной структурой, представляющей собой последовательность записей, содержащих по два поля каждая: поле ключа строкового типа и поле данных типа int. Табличную структуру реализовать в виде объекта, снабдив его методами включения новой записи и выборки записи по заданному ключу. При этом необходимо соблюдать условие уникальности каждого ключа.

#### 2. Описание входных и выходных данных

```
class TTable
{
    //fields
    //current count of records
    int C_Curr_Rec;
    //key-fields
    char *Key[C_Max_Rec+1];
    //data-fields
    int Data[C_Max_Rec+1];

public:
    //methods
    //procedure of initialization
    void Initialization();
    //adds new element (returns "false" if it is impossible to do)
    bool AddNewElement(char *AddKey, int AddData);
    //gets data of key (returns -1 if there is not record with this key-field)
    int GetData(char *SearchKey);
    //outputs table
    void Output();
    //deletes table
    void Delete();
};
```

Исходные данные:

TTable NovaTable - таблица;  
char \*AddKey - поле ключа записи, которую необходимо добавить;  
int AddData - поле данных записи, которую необходимо добавить;  
char \*SearchKey - аргумент поиска (ключ, по которому необходимо найти данные).

Выходные данные:



TTable NovaTable - таблица;  
int SearchData - поле данных записи с полем ключа SearchKey.

### 3. Набор тестов

#### Исходные данные:

NovaTable:

C 20

E 40

S 10

A 40

F 30;

1)

AddKey="B";

AddData=50;

SearchKey="S".

2)

AddKey="E";

AddData=30;

SearchKey="D".

#### Выходные данные:

1)

NovaTable:

C 20

E 40

S 10

A 40

F 30

B 50;

SearchData=10.

2)

NovaTable:

C 20

E 30

S 10

A 40

F 30;

SearchData=-1

(сообщение на экране: "There is not record with this key in the table").

### 4. Описание алгоритма

#### 1. Метод включения новой записи.

Сначала производится последовательный просмотр всех записей таблицы. Если поле ключа какой-либо ранее созданной записи совпадает с полем ключа записи, которую необходимо добавить в таблицу, то поле данных старой записи меняется на поле данных новой записи, и увеличения длины таблицы не происходит. Таким образом соблюдается условие уникальности каждого ключа. Если же совпадения не произошло, и текущее количество записей меньше максимального, то новая запись добавляется в конец таблицы. Иначе добавления не происходит, о чём уведомляет сообщение на экране.

#### 2. Метод выборки записи по заданному ключу.

Производится последовательный просмотр записей таблицы. Если поле ключа какой-либо записи совпадает с аргументом поиска, то считывается поле данных этой записи. Если же записи с таким полем ключа нет, то об этом уведомляет сообщение на экране.

## 5. Текст программы

```
#include <iostream.h>
#include <string.h>
#include <malloc.h>

//max count of records
const C_Max_Rec=255;

//new class
class TTable
{
    //fields
    //current count of records
    int C_Curr_Rec;
    //key-fields
    char *Key[C_Max_Rec+1];
    //data-fields
    int Data[C_Max_Rec+1];

public:
    //methods
    //procedure of initialization
    void Initialization();
    //add new element (return "false" if it is impossible to do)
    bool AddNewElement(char *AddKey, int AddData);
    //get data of key
    int GetData(char *SearchKey);
    //output table
    void Output();
    //delete table
    void Delete();
};

void TTable::Initialization()
{
    //current count of records is zero
    C_Curr_Rec=0;
    //taking memory for key-field of 0-record of table
    Key[0]=(char*) malloc(255);
    //program does not use 0-record of table
    strcpy(Key[0], "Key");
    Data[0]=0;
}

bool TTable::AddNewElement(char *AddKey, int AddData)
{
    //parameter of cycle
    int i;
    //if current count of records and max count of records
    //are equivalent then it is impossible to add new element
    if (C_Curr_Rec==C_Max_Rec)
    {
        //error of adding
        return false;
    }
    //if key which is wanted to add and one of old keys are
    //equivalent then data of old key is changing on new data
    for (i=1; i<=C_Curr_Rec; i++)
    {
```

```

        if (strcmp(Key[i],AddKey)==0)
        {
            strcpy(Key[i], AddKey);
            Data[i]=AddData;
            //adding is successful
            return true;
        }
    }
    //current count of records is increased
    C_Curr_Rec++;
    //taking memory for key-field of new record of table
    Key[C_Curr_Rec]=(char*) malloc(255);
    //new record is added in the end of table
    strcpy(Key[C_Curr_Rec], AddKey);
    Data[C_Curr_Rec]=AddData;
    //adding is successful
    return true;
}

int TTable::GetData(char *SearchKey)
{
    //parameter of cycle
    int i;
    //if key is found then data of this key is returned
    for (i=1; i<=C_Curr_Rec; i++)
    {
        if (strcmp(Key[i],SearchKey)==0)
        {
            //search is successful
            return Data[i];
        }
    }
    //error of search
    return -1;
}

void TTable::Output()
{
    //parameter of cycle
    int i;
    //each record is outputted
    cout<<"The table is:"<<"\n";
    for (i=1; i<=C_Curr_Rec; i++) cout<<Key[i]<<" : "<<Data[i]<<"\n";
}

void TTable::Delete()
{
    //parameter of cycle
    int i;
    //each key-field is deleted
    for (i=0; i<=C_Curr_Rec; i++) free(Key[i]);
}

void main()
{
    //table which will be operated on
    TTable NovaTable;
    //action which will be done
    int Action;
    //data-field of new record
    int AddData;
    //key-field of new record
    char *AddKey;
    //argument of search
    char *SearchKey;
    //data of argument of search
    int SearchData;

```

```

//initialization of table
NovaTable.Initialization();
//shell
do
{
    //outputing of menu
    cout<<"What do you want?"<<"\n";
    cout<<"1. Add new record"<<"\n";
    cout<<"2. Find data"<<"\n";
    cout<<"3. Output table"<<"\n";
    cout<<"4. Exit"<<"\n";
    cin>>Action;
    //adding of new record
    if (Action==1)
    {
        //initialization AddKey
        AddKey=(char*) malloc(255);
        //inputing of key-field of new record
        cout<<"Input new key"<<"\n";
        cin>>AddKey;
        //inputing of data-field of new record
        cout<<"Input new data"<<"\n";
        cin>>AddData;
        //adding of new record
        if ((NovaTable.AddNewElement(AddKey,AddData))==false)
        {
            //if error of adding
            cout<<"It is impossible to add new record!"<<"\n";
        }
        //deinitialization of AddKey
        free(AddKey);
    }
    //finding data
    if (Action==2)
    {
        //initialization of SearchKey
        SearchKey=(char*) malloc(255);
        //inputing of argument of search
        cout<<"Input argument of search"<<"\n";
        cin>>SearchKey;
        //outputing of result of search
        SearchData=NovaTable.GetData(SearchKey);
        if (SearchData!=-1)
        {
            cout<<"The data of this key is:"<<"\n";
            cout<<SearchData<<"\n";
        }
        else
        {
            cout<<"There is not record with this key in the table"<<"\n";
        }
        //deinitialization of SearchKey
        free(SearchKey);
    }
    //outputing of table
    if (Action==3) NovaTable.Output();
}
while (Action!=4);

//deleting of table
NovaTable.Delete();
}

```

## 6. Анализ результатов и выводы

Программа корректно работает и выполняет поставленную задачу. Объём программы составил 180269 байт. При включении новой записи в таблицу с  $n$  записями производится  $n$  операций просмотра поля ключа записи и одна операция добавления новой записи. При замене старого поля данных записи на новое в среднем производится  $(n+1)/2$  операций просмотра поля ключа записи и одна операция замены поля данных записи. При выборке записи по заданному ключу в среднем производится  $(n+1)/2$  операций просмотра поля ключа записи.

## ЛАБОРАТОРНАЯ РАБОТА № 9

### И снова многословные операнды

#### 1. Постановка задачи

Разработать объекты-числа, значения которых представляются массивами байтов, предусмотрев возможность выполнения над ними следующих операций: арифметическое сложение, логическое побитовое сложение, логическое побитовое умножение, сдвиг влево на разряд, сдвиг вправо на разряд. В числе 6 байт. Нулевой байт содержит код знака числа (0 – "+", 255 – "-"), остальные – значащие цифры числа. Задание выполнить в среде Visual C++ в виде консольной программы.

#### 2. Описание входных и выходных данных

```
//new types
typedef unsigned char TContents;
typedef bool          TOverflow;
typedef bool          TDownflow;

//count of bytes in operand
const C_B=6;

//operand
struct TOperand
{
    //contents of operand
    TContents Contents[C_B];
    //overflow
    TOverflow Overflow;
    //downflow :)
    TDownflow Downflow;
};

//new class
class TCalc
{
private:
    //operand
    TOperand Operand;
    //inverting of operand
    void Inv_O();
public:
    //setting of operand
    void Set_O(TContents NewContents[C_B], TOverflow NewOverflow, TDownflow
                NewDownflow);

    //clearing of operand
```

```

void Nil_O();
//inputing of operand
void Inp_O();
//doing of arithmetical addition
void A_Add(TCalc O1, TCalc O2);
//doing of logical addition
void L_Add(TCalc O1, TCalc O2);
//doing of logical multiplication
void L_Mul(TCalc O1, TCalc O2);
//doing of right shift
void S_Rig();
//doing of left shift
void S_Lef();
//outputing of operand
void Out_O();
};

```

### Исходные данные:

#### 1. Арифметическое сложение:

TCalc O1 - первый операнд;

TCalc O2 - второй операнд.

#### 2. Логическое побитовое сложение:

TCalc O1 - первый операнд;

TCalc O2 - второй операнд.

#### 3. Логическое побитовое умножение:

TCalc O1 - первый операнд;

TCalc O2 - второй операнд.

#### 4. Сдвиг вправо:

TCalc O3 - операнд до сдвига.

#### 5. Сдвиг влево:

TCalc O3 - операнд до сдвига.

### Выходные данные:

#### 1. Арифметическое сложение:

TCalc O3 - результат.

#### 2. Логическое побитовое сложение:

TCalc O3 - результат.

#### 3. Логическое побитовое умножение:

TCalc O3 - результат.

#### 4. Сдвиг вправо:

TCalc O3 - операнд после сдвига.

#### 5. Сдвиг влево:

TCalc 03 – операнд после сдвига.

### 3. Набор тестов

#### Исходные данные:

##### 1. Арифметическое сложение:

1)

01.Operand.Contents = 000 231 123 123 001 019;

02.Operand.Contents = 255 123 000 000 003 018.

2)

01.Operand.Contents = 255 231 123 123 001 049;

02.Operand.Contents = 255 123 000 000 003 218.

##### 2. Логическое побитовое сложение:

01.Operand.Contents = 000 231 123 123 001 019;

02.Operand.Contents = 255 123 000 000 003 018.

##### 3. Логическое побитовое умножение:

01.Operand.Contents = 000 231 123 123 001 019;

02.Operand.Contents = 255 123 000 000 003 018.

##### 4. Сдвиг вправо:

03.Operand.Contents = 000 255 127 000 122 017.

##### 5. Сдвиг влево:

1)

03.Operand.Contents = 000 069 127 001 127 128.

2)

03.Operand.Contents = 255 169 127 001 127 128.

#### Выходные данные:

##### 1. Арифметическое сложение:

1)

03.Operand.Contents = 000 108 123 122 254 001;

03.Operand.Overflow = 0.

2)

03.Operand.Contents = 255 098 123 123 005 011;

03.Operand.Overflow = 1.

##### 2. Логическое побитовое сложение:

03.Operand.Contents = 255 255 123 123 003 019.

##### 3. Логическое побитовое умножение:

03.Operand.Contents = 000 099 000 000 001 018.

##### 4. Сдвиг вправо:

03.Operand.Contents = 000 127 191 128 061 008;

03.Operand.Downflow = 1.

##### 5. Сдвиг влево:

1)

03.Operand.Contents = 000 138 254 002 255 000;

```
O3.Operand.Overflow = 0.
```

2)

```
O3.Operand.Contents = 255 082 254 002 255 000;
```

```
O3.Operand.Overflow = 1.
```

#### 4. Описание алгоритма

##### 1) Арифметическое сложение

Непосредственное сложение модулей полей Contents операндов осуществляется путём одновременного этих полей (кроме знаковых байтов) справа налево, при котором соответствующие байты складываются. Если результат сложения больше 255, то в соответствующий байт поля Contents результата записывается [результат сложения – 256], а к результату сложения следующих байтов добавляется 1. Если следующих байтов нет, то в поле Overflow результата остаётся 1.

Если оба операнда положительны (отрицательны), то их модули просто складываются. При этом знаком результата становится "+" ("–").

Если операнды имеют разные знаки, то происходит сложение модуля инверсного кода поля Contents отрицательного операнда с модулем поля Contents положительного операнда. Если произошло переполнение, то модуль поля Contents результата инвертируется, а его знаком становится "–". Иначе его знаком становится "+". В любом случае поле Overflow результата обнуляется.

Получение инверсного кода модуля поля Contents операнда происходит путём просмотра этого поля справа налево (кроме знакового байта). При этом первый ненулевой байт принимает значение [256 – текущее значение], а следующие байты принимают значения [255 – текущее значение].

##### 2) Логическое сложение

Поля Contents операндов одновременно просматриваются справа налево, и к их соответствующим байтам применяется операция "|" (логическое ИЛИ).

##### 3) Логическое умножение

Поля Contents операндов одновременно просматриваются справа налево, и к их соответствующим байтам применяется операция "&" (логическое И).

##### 4) Сдвиг вправо на разряд

Поле Contents операнда (кроме знакового байта) просматривается слева направо, и каждый байт делится на 2 с отбрасыванием остатка. Если этот остаток равен 1, то к следующему байту после его деления на 2 добавляется 128. Если следующего байта нет, то в поле Downflow оператора остаётся 1.

##### 5) Сдвиг влево на разряд

Поле Contents операнда (кроме знакового байта) просматривается справа налево, и каждый байт умножается на 2. Если результат умножения больше 255, то в текущий байт записывается [результат умножения – 256], а к следующему байту после его умножения на 2 добавляется 1. Если следующего байта нет, то в поле Overflow оператора остаётся 1.

#### 5. Текст программы

```
#include <iostream.h>
```

```
//new types
```

```
typedef unsigned char TContents;
```

```
typedef bool TOverflow;
```



```

typedef bool TDownflow;

//count of bytes in operand
const C_B=6;

//operand
struct TOperand
{
    //contents of operand
    TContents Contents[C_B];
    //overflow
    TOverflow Overflow;
    //downflow :)
    TDownflow Downflow;
};

//new class
class TCalc
{
private:
    //operand
    TOperand Operand;
    //inverting of operand
    void Inv_O();
public:
    //setting of operand
    void Set_O(TContents NewContents[C_B], TOverflow NewOverflow, TDownflow
        NewDownflow);

    //clearing of operand
    void Nil_O();
    //inputing of operand
    void Inp_O();
    //doing of arithmetical addition
    void A_Add(TCalc O1, TCalc O2);
    //doing of logical addition
    void L_Add(TCalc O1, TCalc O2);
    //doing of logical multiplication
    void L_Mul(TCalc O1, TCalc O2);
    //doing of right shift
    void S_Rig();
    //doing of left shift
    void S_Lef();
    //outputing of operand
    void Out_O();
};

void TCalc::Inv_O()
{
    //parameter of cycle
    short i;
    //find first not zero-element
    i=C_B-1;
    while ((Operand.Contents[i]==0) & (i>=1)) i--;
    //inverting of all bytes of contents and adding 1 to the first not zero-element
    if (i>=1)
    {
        Operand.Contents[i]=256-Operand.Contents[i];
        for (i=i-1; i>=1; i--) Operand.Contents[i]=255-Operand.Contents[i];
    }
}

void TCalc::Set_O(TContents NewContents[C_B], TOverflow NewOverflow, TDownflow
NewDownflow)
{
    //parameter of cycle
    short i;
    //setting of new contents

```

```

    for (i=0; i<C_B; i++) Operand.Contents[i]=NewContents[i];
    //setting of new overflow
    Operand.Overflow=NewOverflow;
    //setting of new downflow
    Operand.Downflow=NewDownflow;
}

void TCalc::Nil_O()
{
    //0-contents
    TContents NilContents[C_B]={0,0,0,0,0,0};
    //setting
    Set_O(NilContents,0,0);
}

void TCalc::Inp_O()
{
    //parameter of cycle
    short i;
    //new contents (char)
    TContents NewContents[C_B];
    //new contents (unsigned short) for inputing
    unsigned short NewContents1[C_B];
    //inputing of new contents
    for (i=0; i<=C_B-1; i++)
    {
        cout<<"Input new "<<i<<" byte -> ";
        cin>>NewContents1[i];
    }
    //transforming unsigned short into char
    for (i=0; i<=C_B-1; i++) NewContents[i]= (char) NewContents1[i];
    //setting
    Set_O(NewContents, 0, 0);
}

void TCalc::A_Add(TCalc O1, TCalc O2)
{
    //parameter of cycle
    short i;
    //buffer
    short buf;
    //if only one minus-operand then this operand is inverted
    if ((O1.Operand.Contents[0]==255) ^ (O2.Operand.Contents[0]==255))
    {
        if (O1.Operand.Contents[0]==255) O1.Inv_O();
        if (O2.Operand.Contents[0]==255) O2.Inv_O();
    }
    //if two minus-operands then sign of result is minus
    if ((O1.Operand.Contents[0]==255) & (O2.Operand.Contents[0]==255))
    Operand.Contents[0]=255;
    //doing of sequence of additions with moving of overflow on the left
    for (i=C_B-1; i>=1; i--)
    {
        buf=O1.Operand.Contents[i]+O2.Operand.Contents[i]+Operand.Overflow;
        Operand.Overflow=(buf>255);
        if (Operand.Overflow==1) Operand.Contents[i]=buf-256;
        else Operand.Contents[i]=buf;
    }
    //if only one minus-element
    if ((O1.Operand.Contents[0]==255) ^ (O2.Operand.Contents[0]==255))
    {
        //...and overflow of result is 0 then result is inverted and sign of result
        //is minus
        if (Operand.Overflow==0)
        {
            Inv_O();
            Operand.Contents[0]=255;
        }
    }
}

```

```

    }
    //overflow of result is 0
    Operand.Overflow=0;
}
}

void TCalc::L_Add(TCalc O1, TCalc O2)
{
    //parameter of cycle
    short i;
    //doing sequence of logical additions
    for (i=C_B-1; i>=0; i--)
        Operand.Contents[i]=(O1.Operand.Contents[i])|(O2.Operand.Contents[i]);
}

void TCalc::L_Mul(TCalc O1, TCalc O2)
{
    //parameter of cycle
    short i;
    //doing of sequence of logical multiplications
    for (i=C_B-1; i>=0; i--)
        Operand.Contents[i]=(O1.Operand.Contents[i])&(O2.Operand.Contents[i]);
}

void TCalc::S_Rig()
{
    //parameter of cycle
    short i;
    //buffer
    short buf;
    //doing of sequence of divisions by 2 with moving of the downflow (rest from
    //division) on the right
    for (i=1; i<=C_B-1; i++)
    {
        buf=Operand.Contents[i]/2+128*Operand.Downflow;
        if (Operand.Contents[i]%2==1) Operand.Downflow=1;
        else Operand.Downflow=0;
        Operand.Contents[i]=buf;
    }
}

void TCalc::S_Lef()
{
    //parameter of cycle
    short i;
    //buffer
    short buf;
    //doing of sequence of doubling with left moving of the overflow on the left
    for (i=C_B-1; i>=1; i--)
    {
        buf=Operand.Contents[i]*2+Operand.Overflow;
        if (Operand.Contents[i]>127)
        {
            Operand.Overflow=1;
            buf=buf-256;
        }
        else Operand.Overflow=0;
        Operand.Contents[i]=buf;
    }
}

void TCalc::Out_O()
{
    //parameter of cycle
    short i;
    //contents for outputing
    unsigned short Contents1[C_B];

```

```

//transforming char into unsigned short
for (i=0; i<=C_B-1; i++) Contents1[i]=(unsigned short) Operand.Contents[i];
//outputing of contents
cout<<"Contents: ";
for (i=0; i<=C_B-1; i++) cout<<Contents1[i]<<" ";
cout<<"\n";
//outputing of overflow
cout<<"Overflow: "<<Operand.Overflow<<"\n";
//outputing of downflow
cout<<"Downflow: "<<Operand.Downflow<<"\n";
}

void main()
{
    //objects which will be operated on
    TCalc O1,O2,O3;
    //action
    short Act;
    //shell
    do
    {
        //output menu
        cout<<"What do you want?"<<"\n";
        cout<<"1. Do arithmetical addition"<<"\n";
        cout<<"2. Do logical addition"<<"\n";
        cout<<"3. Do logical multiplication"<<"\n";
        cout<<"4. Do right shift"<<"\n";
        cout<<"5. Do left shift"<<"\n";
        cout<<"6. Exit"<<"\n";
        //input action
        cin>>Act;
        //do arithmetical addition
        if (Act==1)
        {
            O1.Nil_O();
            cout<<"Input first operand:"<<"\n";
            O1.Inp_O();
            O2.Nil_O();
            cout<<"Input second operand:"<<"\n";
            O2.Inp_O();
            O3.Nil_O();
            O3.A_Add(O1,O2);
            cout<<"The result is:"<<"\n";
            O3.Out_O();
        }
        //do logical addition
        if (Act==2)
        {
            O1.Nil_O();
            cout<<"Input first operand:"<<"\n";
            O1.Inp_O();
            O2.Nil_O();
            cout<<"Input second operand:"<<"\n";
            O2.Inp_O();
            O3.Nil_O();
            O3.L_Add(O1,O2);
            cout<<"The result is:"<<"\n";
            O3.Out_O();
        }
        //do logical multiplication
        if (Act==3)
        {
            O1.Nil_O();
            cout<<"Input first operand:"<<"\n";
            O1.Inp_O();
            O2.Nil_O();
            cout<<"Input second operand:"<<"\n";

```

```

        O2.Inp_O();
        O3.Nil_O();
        O3.L_Mul(O1,O2);
        cout<<"The result is:"<<"\n";
        O3.Out_O();
    }
    //do right shift
    if (Act==4)
    {
        O3.Nil_O();
        cout<<"Input the operand:"<<"\n";
        O3.Inp_O();
        O3.S_Rig();
        cout<<"The result is:"<<"\n";
        O3.Out_O();
    }
    //do left shift
    if (Act==5)
    {
        O3.Nil_O();
        cout<<"Input the operand:"<<"\n";
        O3.Inp_O();
        O3.S_Lef();
        cout<<"The result is:"<<"\n";
        O3.Out_O();
    }
}
//until exit
while (Act!=6);
}

```

## 6. Анализ результатов и выводы

Программа корректно работает и выполняет поставленную задачу. Объём программы составил 196656 байт. Операции получения инверсного кода, арифметического сложения операндов с одинаковыми знаками, логического сложения, логического умножения, сдвига влево и сдвига вправо осуществляются за один полный просмотр поля Contents операнда(ов). Операция сложения разнознаковых операндов осуществляется за 1-2 операции инвертирования и один полный просмотр полей Contents операндов.

## ЛАБОРАТОРНАЯ РАБОТА № 10

### Диалоговое окно

#### 1. Постановка задачи

В проекте однодокументной (SDI) программы создать диалоговое окно с двумя текстовыми полями, одно из которых предназначено для ввода двоичного числа, а второе — для вывода его шестнадцатиричного эквивалента. Выполнение преобразования связать с нажатием специально предусмотренной для этой цели кнопки диалогового окна. После закрытия диалогового окна кнопкой «ОК» обе строки следует вывести в клиентскую область главного окна. Запуск диалогового окна связать с пунктом меню главного окна.

## 2. Описание входных и выходных данных

Исходные данные:

CString m\_BinEdit - строка, содержащая изображение двоичного числа.

Выходные данные:

CString m\_HexEdit - строка, содержащая изображение его шестнадцатеричного эквивалента.

## 3. Набор тестов

Исходные данные:

```
1)
m_BinEdit = "10001011";
2)
m_BinEdit = "1010110010";
3)
m_BinEdit = "10012010".
```

Выходные данные:

```
1)
m_HexEdit = "8B";
2)
m_HexEdit = "2B2";
3)
m_HexEdit = "Wrong input!".
```

## 4. Описание алгоритма

Сначала m\_BinEdit проверяется на принадлежность каждого её символа алфавиту двоичной системы счисления. В случае ошибки в m\_HexEdit записывается "Wrong input!". Иначе выполняется перевод в шестнадцатеричную систему счисления. M\_BinEdit просматривается с конца и каждая четвёрка цифр из неё ассоциируется с одной цифрой шестнадцатеричной системы счисления. Результаты перевода четвёрок дописываются в начало m\_HexEdit. Так продолжается до тех пор, пока не будет достигнуто начало строки. Если четвёрка получилась неполной, то считается, что незанятые символы в её начале – ноли.

## 5. Текст программы

```
...

//translate 4 binary digits in 1 hex digit
char GoToHex(char s[4])
{
    int a;

    a=atoi(s);
    switch(a)
    {
        case 0: return '0';
        case 1: return '1';
        case 10: return '2';
        case 11: return '3';
        case 100: return '4';
        case 101: return '5';
```

```

        case 110: return '6';
        case 111: return '7';
        case 1000: return '8';
        case 1001: return '9';
        case 1010: return 'A';
        case 1011: return 'B';
        case 1100: return 'C';
        case 1101: return 'D';
        case 1110: return 'E';
        case 1111: return 'F';
        default : return -1 ;
    }
}

//find input error
int GetError(CString m_BinEdit)
{
    int i;

    for (i=strlen(m_BinEdit)-1; i>=0; i--)
        if (m_BinEdit[i]!='0') & (m_BinEdit[i]!='1')) return -1;
    return 0;
}

//update&translate&update
void TDialog::OnButton1()
{
    //4 binary digits
    char s[4];
    int i,j=0;

    UpdateData(true);
    m_HexEdit="";
    if (GetError(m_BinEdit)==0)
    {
        for (i=strlen(m_BinEdit)-1; i>=0; i--)
        {
            j=j%4;
            s[3-j]=m_BinEdit[i];
            if ((j==3) | (i==0))
            {
                m_HexEdit=GoToHex(s)+m_HexEdit;
                strcpy(s,"000");
            }
            j++;
        }
    }
    if (GetError(m_BinEdit)==-1) m_HexEdit="Wrong input!";
    UpdateData(false);
}

...

```

## 6. Анализ результатов и выводы

Программа корректно работает и выполняет поставленную задачу. Производится проверка на ошибку ввода. Объём программы составил 122932 байт. При  $n$  символах в `m_BinEdit` проверка на ошибку ввода выполняется за  $n$  операций сравнения. Перевод осуществляется за  $\lfloor n/4 \rfloor$  операций перевода одной четвёрки. Операция перевода одной четвёрки в среднем выполняется за  $17/2$  операций сравнения.

1. Вычислим  $r$  - остаток от деления числа  $a$  на  $b$ ,  $a = bq + r$ ,  $0 \leq r < b$ .
2. Если  $r = 0$ , то  $b$  есть искомое число.
3. Если  $r \neq 0$ , то заменим пару чисел  $(a, b)$  парой  $(b, r)$  и перейдем к шагу 1.

```
int NOD(int a, int b)
{
    while (a != 0 && b != 0)
    {
        if (a >= b) a = a % b;
        else b = b % a;
    }
    return a + b; // Одно - ноль
}
```

При вычислении наибольшего общего делителя  $(a, b)$  с помощью алгоритма Евклида будет выполнено не более  $5r$  операций деления с остатком, где  $r$  есть количество цифр в десятичной записи меньшего из чисел  $a$  и  $b$ .

### Бинарный алгоритм Евклида

Этот алгоритм использует соотношения для НОД:

$$\begin{aligned} \text{НОД}(2*a, 2*b) &= 2*\text{НОД}(a, b) \\ \text{НОД}(2*a, b) &= \text{НОД}(a, b) \text{ при нечетном } b, \end{aligned}$$

Он иллюстрируется следующей программой:

```
m:= a; n:=b; d:=1;
{НОД(a,b) = d * НОД(m,n)}
while not ((m=0) or (n=0)) do begin
    if (m mod 2 = 0) and (n mod 2 = 0) then begin
        d:= d*2; m:= m div 2; n:= n div 2;
    end else if (m mod 2 = 0) and (n mod 2 = 1) then begin
        m:= m div 2;
    end else if (m mod 2 = 1) and (n mod 2 = 0) then begin
        n:= n div 2;
    end else if (m mod 2=1) and (n mod 2=1) and (m>=n) then begin
        m:= m-n;
    end else if (m mod 2=1) and (n mod 2=1) and (m<=n) then begin
        n:= n-m;
    end;
end;
{m=0 => ответ=d*n; n=0 => ответ=d*m}
```

### Алгоритм решения уравнения $ax+by = 1$



1. Определим матрицу E:

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

2. Вычислим  $r$  - остаток от деления числа  $a$  на  $b$ ,  $a = bq + r$ ,  $0 \leq r < b$ .

3. Если  $r = 0$ , то второй столбец матрицы E даёт вектор  $(x, y)$  решений уравнения.

4. Если  $r \neq 0$ , то заменим матрицу E матрицей

$$E * \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix}$$

5. Заменим пару чисел  $(a, b)$  на  $(b, r)$  и перейдем к шагу 2.

## Расширенный алгоритм Евклида

Алгоритм Евклида можно расширить так, что он не только даст  $\text{НОД}(a, b) = d$ , но и найдет целые числа  $x$  и  $y$ , такие что  $ax + by = d$ .

### Псевдокод.

НА ВХОДЕ: два неотрицательных числа  $a$  и  $b$ :  $a \geq b$

НА ВЫХОДЕ:  $d = \text{НОД}(a, b)$  и целые  $x, y$ :  $ax + by = d$ .

1. Если  $b = 0$  положить  $d := a$ ,  $x := 1$ ,  $y := 0$  и вернуть  $(d, x, y)$
2. Положить  $x_2 := 1$ ,  $x_1 := 0$ ,  $y_2 := 0$ ,  $y_1 := 1$
3. Пока  $b > 0$ 
  - 3.1  $q := \lfloor a/b \rfloor$ ,  $r := a - qb$ ,  $x := x_2 - qx_1$ ,  $y := y_2 - qy_1$
  - 3.2  $a := b$ ,  $b := r$ ,  $x_2 := x_1$ ,  $x_1 := x$ ,  $y_2 := y_1$ ,  $y_1 := y$
4. Положить  $d := a$ ,  $x := x_2$ ,  $y := y_2$  и вернуть  $(d, x, y)$

### Исходник на Си.

```
/* Author:  Pate Williams (c) 1997 */

#include <stdio.h>

#define DEBUG

void extended_euclid(long a, long b, long *x, long *y, long *d)
/* calculates a * *x + b * *y = gcd(a, b) = *d */
{
    long q, r, x1, x2, y1, y2;

    if (b == 0) {
        *d = a, *x = 1, *y = 0;
```

```

    return;

}

x2 = 1, x1 = 0, y2 = 0, y1 = 1;

#ifdef DEBUG
printf("-----");
printf("-----\n");
printf("q      r      x      y      a      b      ");
printf("x2      x1      y2      y1\n");
printf("-----");
printf("-----\n");
#endif

while (b > 0) {

    q = a / b, r = a - q * b;

    *x = x2 - q * x1, *y = y2 - q * y1;

    a = b, b = r;

    x2 = x1, x1 = *x, y2 = y1, y1 = *y;

#ifdef DEBUG
printf("%4ld %4ld %4ld %4ld ", q, r, *x, *y);
printf("%4ld %4ld %4ld %4ld ", a, b, x2, x1);
printf("%4ld %4ld\n", y2, y1);
#endif

}

*d = a, *x = x2, *y = y2;

#ifdef DEBUG
printf("-----");
printf("-----\n");
#endif

}

int main(void)
{

    long a = 4864, b = 3458, d, x, y;

    extended_euclid(a, b, &x, &y, &d);

    printf("x = %ld y = %ld d = %ld\n", x, y, d);

    return 0;

}

```

Алгоритм работает за  $O(\log^2 n)$  операций.

## НОК

$$\text{НОК}(a, b) = a * b / \text{НОД}(a, b)$$

## Проверка, високосный год или нет

1. Год делится на 400 -> високосный -> конец
2. Год делится на 100, но не делится на 400 -> не високосный -> конец
3. Год делится на 4, но не делится на 100 -> високосный -> конец
4. Год не високосный -> конец
5. Всё

Задано целое положительное число  $n$ . Разработать программу вычисления значения  $y$  по формуле:  $y = (1 + (2 + (3 + \dots (n - 1 + n^{1/2})^{1/2} \dots)^{1/2})^{1/2})^{1/2}$ .

Разработать программу вычисления  $n$  чисел последовательности (ряда) Фибоначчи. Последовательность Фибоначчи задается множеством чисел, каждое из которых определяется как сумма двух предыдущих, за исключением первых двух элементов последовательности - они равны единице:  $F_0 = F_1 = 1$ ;  $F_n = F_{n-1} + F_{n-2}$ .

Разработать программу транспонирования матрицы случайных чисел размерности  $3 \times 3$ .

Разработать программу, которая определяет цифровой корень числа. Цифровой корень числа находится через сумму составляющих его цифр до тех пор, пока эта сумма сама не станет цифрой. Например, для числа 155 цифровой корень находится так:  $1 + 5 + 5 = 11$ ,  $1 + 1 = 2$ . Цифровой корень 155 равен 2.

## Линейные программы

1. Вычислить произведение высот треугольника со сторонами **a**, **b**, **c**.
2. Вычислить длину высоты, опущенную на гипотенузу прямоугольного треугольника с катетами **a** и **b**.
3. Вычислить площадь поверхности и объем правильной пирамиды, в основании которой квадрат со стороной **a** и высота **h**.
4. Написать программу, которая вводит два вещественных числа, вычисляет и печатает коэффициенты приведенного квадратного уравнения, корнями которого являются эти числа.

5. Треугольник задан координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  своих вершин. Вычислить радиус окружности, вписанной в треугольник.
6. Ромб задан координатами трех вершин  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Вычислить площадь и периметр ромба.
7. Вычислить время падения тела с высоты  $H$  с начальной скоростью  $V$ .
8. Вычислить площадь треугольника со сторонами  $k$  и прилежащими углами  $a$  и  $b$ .
9. Смешаны  $V_1$  литр воды температуры  $T_1$  с  $V_2$  литрами воды температуры  $T_2$ . Написать программу вычисления объема и температуры воды.
10. Треугольник задан координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  своих вершин. Найти периметр и площадь треугольника.
11. Найти площадь кольца, внутренний радиус которого равен 20, внешний - заданному числу  $R > 20$ .
12. Известно, что точки с координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  являются тремя вершинами некоторого параллелограмма. Найти координаты четвертой вершины.
13. Вычислить длину окружности, площадь круга и объем шара одного радиуса.
14. По длинам двух сторон треугольника и углу между ними найти длину третьей стороны и площадь треугольника.
15. Треугольник задан координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  своих вершин. Вычислить радиус окружности, описанной около треугольника.

### Логические выражения

Написать на языке С логическое выражение, принимающее значение **истина** или **ложь** при выполнении условия:

1. Треугольник с вершинами  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$  является равносторонним.
2. Прямая, проходящая через точки  $A(x_1, y_1)$  и  $B(x_2, y_2)$  параллельна прямой, проходящей через точки  $C(x_3, y_3)$  и  $D(x_4, y_4)$ .
3. Отрезок, соединяющий точки  $A(x_1, y_1)$  и  $B(x_2, y_2)$  перпендикулярен отрезку, соединяющему точки  $C(x_3, y_3)$  и  $D(x_4, y_4)$ .
4. Точка  $M(x, y)$  принадлежит верхней части единичного круга с центром в точке  $O(a, b)$ .
5. Ветви параболы  $y = ax^2 + bx + c$  направлены вверх, а корни являются взаимно противоположными числами.
6. Цилиндр, радиус основания которого равен  $R$ , проходит через треугольное отверстие со сторонами  $a$ ,  $b$ ,  $c$ .
7. Точка  $M(a, b)$  лежит на биссектрисе 2-го и 4-го координатных углов и находится внутри единичного круга с центром в начале координат.

8. Натуральное число  $p$  делится нацело на натуральное число  $q$ .
9. Треугольник с вершинами  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$  является равносторонним.
10. Прямоугольный параллелепипед с ребрами  $a$ ,  $b$  и  $c$  проходит через прямоугольное отверстие со сторонами  $m$  и  $n$ .
11. Точка  $M(x, y)$  принадлежит прямоугольнику, ограниченному прямыми  $x=2$ ,  $y=1$  и координатными осями.
12. Точка  $M(x, y)$  принадлежит области, ограниченной левой частью единичной окружности и осью  $y$ .
13. Круг радиуса  $R$  поместится в квадрат со стороной  $a$ .

## Ветвления

### Задание 1

1. Найти  $\max(a, b)$ .
2. Вычислить  $y(x)$ , если  $y = x^2$  при  $x > 1$  и  $y = x$  при  $x \leq 1$ .
3. Даны два отрезка  $[a, b]$ ,  $[c, d]$  на прямой. Установить, имеют ли они общие точки или нет.
4. Дана точка  $M(x, y)$ . Присвоить  $z = 1$ , если точка принадлежит окружности с радиусом  $R$  и центром в точке  $(a, b)$  и  $z = 0$  в противном случае.
5. Вычислить функцию  $f(x)$ , если  $f(x) = e^{-x}$  при  $x \geq 0$  и  $f(x) = \cos x$  при  $x < 0$ .
6. Составить программу вычисления модуля  $|5x - 4|$ .
7. Составить программу вычисления функции  $y = \lg(3x - 6)$ .
8. Напечатать координаты той из точек  $M_1(a, b)$  и  $M_2(c, d)$ , которая расположена ближе к началу координат.

### Задание 2

1. Даны уравнения прямых  $a_1x + b_1y = c_1$ ,  $a_2x + b_2y = c_2$ ,  $a_3x + b_3y = c_3$ . Выяснить, какие из этих прямых параллельны и указать, если таковых не имеется.
2. В какой четверти координатной плоскости находится точка с координатами  $x, y$  ( $xy < 0$ ).
3. Если сумма двух различных чисел меньше единицы, то меньшее заменить полусуммой, в противном случае меньшее заменить суммой.
4. Даны различные действительные числа  $x, y, z, d$ . Найти  $\max(\max(x, y), \max(x, z), \max(z, d))$ .
5. Даны три действительных числа. Определить, что больше, сумма или произведение этих чисел. Если сумма больше произведения на число меньше единицы, то вывести  $0$ , и противном случае вывести  $1$ .
6. Даны отрезки  $[a, b]$  и  $[c, d]$  и точка  $A$  с координатой  $x$ . Определить, принадлежит ли данная точка одному из этих отрезков, обоим или лежит вне их.

7. Даны два действительных числа  $x$  и  $y$ . Если наименьшее из них отрицательно, то заменить его нулем, в противном случае единицей.
8. Определить, существует ли треугольник со сторонами  $a$ ,  $b$ ,  $c$ , и если существует, то является ли он равносторонним, равнобедренным или разносторонним.
9. Даны действительные числа  $x_1$ ,  $x_2$ ,  $x_3$ ,  $y_1$ ,  $y_2$ ,  $y_3$ . Принадлежит ли начало координат треугольнику с вершинами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ ?
10. Даны действительные числа  $x$ ,  $y$ . Если  $x$  и  $y$  отрицательные, то  $x$  присвоить модуль  $x$ . Если отрицательное одно из них, то увеличить  $y$  на  $0.5$ . Если оба числа отрицательные, то увеличить  $x$  в  $10$  раз.
11. Даны три действительных числа  $x$ ,  $y$ ,  $z$  и отрезок  $[a, b]$ . Заменить на нули те из них, которые принадлежат отрезку и на единицы - остальные.
12. Даны различные действительные числа  $a$ ,  $b$ ,  $c$ ,  $d$ . Найти  $\max(\max(a, b), \max(a, c), \max(a, d))$ .
13. Известно, что из четырех чисел  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_4$  одно отлично от трех других, равных между собой. Присвоить номер этого числа переменной  $n$ .
14. Даны уравнения прямых  $a_1x + b_1y = c_1$ ,  $a_2x + b_2y = c_2$ ,  $a_3x + b_3y = c_3$ . Выяснить, какие из этих прямых перпендикулярны или указать, если таковых не имеется.

### Задание 3

1. Составить программу определения минимального элемента из трех элементов  $x_1$ ,  $x_2$ ,  $x_3$ .
2. Даны три числа  $a$ ,  $b$ ,  $c$ , удовлетворяющие аксиомам треугольника. Если треугольник равносторонний, то найти его площадь. Если треугольник равнобедренный, то найти периметр и угол между равными сторонами.
3. Даны два числа. Если они не равны, то найти их сумму и произведение. Если произведение больше суммы, то определить на сколько.
4. Определить направление ветвей параболы  $y = ax^2 + by^2 + c$ . Найти точки пересечения параболы с осью  $OX$ .
5. Даны три числа. Определить, существует ли треугольник со сторонами длиной  $a$ ,  $b$ ,  $c$  и, если существует, то найти его параметр и площадь.
6. Вывести сообщение о количестве корней квадратного уравнения и найти эти корни.
7. Если  $w \neq 0$  и, при этом,  $w < 0.5$ , то поменять знак  $w$ , а если  $w = 0$ , присвоить  $w$  единицу.
8. Определить параллельны ли прямые, заданные уравнениями  $y = k_1x + b_1$  и  $y = k_2x + b_2$ . Если они параллельны, то найти координаты точек пересечения с осью  $OX$  ( $k_1$ ,  $k_2 \neq 0$ ).
9. Дана точка  $M(x, y)$ . Проверить, принадлежит ли точка окружности единичного радиуса. Если принадлежит, то уменьшить координату  $x$  на единицу, а  $y$  увеличить на значение  $x$ .

10. Даны две прямые, заданные уравнениями  $y=k_1x+b_1$  и  $y=k_2x+b_2$ . Если эти прямые параллельны, то определить расстояние между ними.
11. Даны прямая и окружность, заданные уравнениями  $y=kx+b$  и  $(x-a)^2+(y-b)^2=r^2$ . Определить, сколько точек пересечения имеют прямая и окружность и найти координаты этих точек.
12. Даны два действительных числа  $a$  и  $b$ . Сравнить их целые части, и если они равны, то поменять местами их дробные части, в противном случае округлить эти числа.
13. Даны две окружности, заданные уравнениями  $(x-a_1)^2+(y-b_1)^2=r_1^2$  и  $(x-a_2)^2+(y-b_2)^2=r_2^2$ . Определить количество точек пересечения и найти их координаты.

### Выбор

1. Найти остаток от деления целой части значения функции  $z = \ln(x^2+ab)$  на 7 и, в зависимости от его величины, напечатать сообщение об одном из дней недели, пронумеровав их от 0 до 6.
2. Даны три числа  $a$ ,  $b$ ,  $c$ , удовлетворяющие аксиоме треугольника и число  $p$ . Выполнить следующие действия: если  $p=1$  - найти периметр треугольника; если  $p=2$  - найти площадь треугольника; если  $p=3$  - найти угол  $a$ . Иначе напечатать слово "треугольник".
3. Найти остаток от деления целого выражения  $a=(c+d)(2k-m)$  на 5 и вывести сообщение о величине остатка. Если остаток равен 0, присвоить  $a$  значение целой части выражения. При нечетном остатке напечатать "нечетное число", при четном - "четное число"
4. Написать программу, которая по номеру машины выводит фамилию студента, сидящего за ней.
5. Найти остаток от деления целой части выражения  $z = \cos(x^2+1)$  на 4 и, в зависимости от его величины, напечатать сообщение об одном из времен года, пронумеровав их от 0 до 3.
6. Найти остаток от деления целого выражения  $c = k(a+b)$  на 4 и вывести сообщение о величине остатка. Если остаток равен 0, то значение переменной оставить без изменения, если 1 или 3 - уменьшить на величину остатка, если 2 - увеличить на величину остатка.
7. По номеру дня недели (1,2,3,4,5,6,7) указать название этого дня. Указать рабочие и выходные дни.
8. Найти остаток от деления целой части выражения  $\sin(a+b)0.5c$  на 4 и вывести на экран сообщение о величине остатка. Если остаток равен 0, то значение выражения заменить на его целую часть. Если остаток равен 1, значение выражения заменить на его дробную часть. В остальных случаях оставить без изменения.
9. В зависимости от времени года "весна", "лето", "осень", "зима" определить погоду "тепло", "жарко", "холодно", "очень холодно".

## Циклы

1. Найти сумму натуральных чисел, предшествующих заданному числу  $a$ .
2. Вычислить сумму квадратов всех целых чисел, меньших заданного числа  $A$ .
3. Дана степень натурального числа  $a$ . Найти ее показатель.
4. Дано **100** вещественных чисел. Вычислить разность между максимальным и минимальным из них.
5. Дана непустая последовательность различных чисел. Определить порядковый номер наименьшего из них.
6. Даны целое  $n > 0$  и последовательность из  $n$  вещественных чисел, среди которых есть хотя бы одно отрицательное число. Найти наибольшее среди отрицательных чисел.
7. Дана последовательность  $a_1, a_2, \dots, a_n$  вещественных чисел. Найти сумму всех элементов, больших заданного числа  $b$ .
8. Дана таблица целых чисел  $a_1, a_2, \dots, a_n$ . Упорядочить эту таблицу.
9. Дана таблица целых чисел  $a_1, a_2, \dots, a_n$ . Найти сумму  $S = a_1 * a_n + a_2 * a_{n-1} + \dots + a_n * a_1$ .
10. Дана таблица целых чисел  $a_1, a_2, \dots, a_n$ . Найти количество чисел, принадлежащих отрезку  $[p, q]$ .
11. Дана таблица целых чисел  $a_1, a_2, \dots, a_n$  и число  $c$ . Найти минимальное значение индекса  $p$ , при котором  $a_1 + a_2 + \dots + a_p < c$ .
12. Дана таблица целых чисел  $a_1, a_2, \dots, a_n$ . Подсчитать количество различных чисел. Составьте новую таблицу, состоящую из различных элементов данной, входящих в нее в такой же последовательности.
13. Даны две последовательности  $a_1 \leq a_2 \leq \dots \leq a_n$  и  $b_1 \leq b_2 \leq \dots \leq b_m$ . Получить таблицу чисел  $c_1 \leq c_2 \leq \dots \leq c_{n+m}$  из эти двух таблиц.
14. Дана таблица целых чисел  $a_1, a_2, \dots, a_n$  и число  $c$ . Расширить эту таблицу до таблицы  $a_1 \leq a_2 \leq \dots \leq a_{n+1}$ , включив в нее число  $c$ .

## Массивы

### Задание 1.

1. Дан  $n$ -мерный вектор  $x = (x_1, x_2, \dots, x_n)$ . Написать программу, которая переставляет компоненты вектора  $x$  так, чтобы в начале стояли положительные, затем нулевые и отрицательные значения в порядке их следования.
2. Даны векторы  $a = (a_1, a_2, a_3)$  и  $b = (b_1, b_2, b_3)$ . Написать программу вычисления скалярного и векторного произведений этих векторов
3. Даны  $m$  векторов  $x_1 = (x_{11}, x_{21}, x_{31}), \dots, x_m = (x_{1m}, x_{2m}, x_{3m})$ . Написать программу нахождения суммы этих векторов.



4. Даны три вектора  $\mathbf{a}=(a_1, a_2, a_3)$ ,  $\mathbf{b}=(b_1, b_2, b_3)$ ,  $\mathbf{c}=(c_1, c_2, c_3)$ . Написать программу вычисления смешанного произведения этих векторов.
5. Даны два вектора  $\mathbf{a}=(a_1, a_2, a_3)$ ,  $\mathbf{b}=(b_1, b_2, b_3)$ . Написать программу, которая находит угол между этими векторами.
6. Даны векторы  $\mathbf{a}=(a_1, a_2, a_3)$ ,  $\mathbf{b}=(b_1, b_2, b_3)$ ,  $\mathbf{c}=(c_1, c_2, c_3)$ ,  $\mathbf{d}=(d_1, d_2, d_3)$ . Написать программу, вычисляющую скалярное произведение  $(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d})$ .
7. Даны две точки в  $n$ -мерном пространстве  $\mathbf{X}=(x_1, x_2, \dots, x_n)$ ,  $\mathbf{Y}=(y_1, y_2, \dots, y_n)$ . Написать программу нахождения расстояния между этими точками и вектора  $\mathbf{XY}$ .
8. Дан  $n$ -мерный вектор  $\mathbf{x}=(x_1, x_2, \dots, x_n)$ . Написать программу, которая может находить вектор  $\mathbf{y}=(x_n, x_{n-1}, x_{n-2}, \dots, x_2, x_1)$  и скалярное произведение  $\mathbf{x} \cdot \mathbf{y}$ .
9. Дан вектор  $\mathbf{a}=(a_1, a_2, a_3)$  и плоскость, заданная уравнением  $Ax + By + Cz = 0$ . Написать программу нахождения угла между векторами и плоскостью.
10. Даны векторы в  $n$ -мерном пространстве  $\mathbf{x}=(x_1, x_2, \dots, x_n)$ ,  $\mathbf{y}=(y_1, y_2, \dots, y_n)$ ,  $\mathbf{z}=(z_1, z_2, \dots, z_n)$ . Написать программу, которая определяет, можно ли из этих векторов построить треугольник и, если можно, найти его площадь.
11. Даны два вектора  $\mathbf{x}=(x_1, x_2, \dots, x_n)$ ,  $\mathbf{y}=(y_1, y_2, \dots, y_n)$ . Написать программу, проверяющую является ли  $\mathbf{x}$  и  $\mathbf{y}$  линейно зависимыми.
12. Даны три вектора  $\mathbf{a}=(a_1, a_2, a_3)$ ,  $\mathbf{b}=(b_1, b_2, b_3)$ ,  $\mathbf{c}=(c_1, c_2, c_3)$ . Написать программу нахождения вектора  $\mathbf{F}=(\mathbf{a} \times \mathbf{b}) \times \mathbf{c}$ .
13. Дан вектор  $\mathbf{x}=(x_1, x_2, \dots, x_n)$ . Написать программу, которая находит разность между первой и последней отрицательными координатами вектора  $\mathbf{x}$ .
14. Дан вектор  $\mathbf{x}=(x_1, x_2, \dots, x_n)$ . Написать программу, которая проверяет, упорядочены ли его координаты по убыванию.
15. Даны три вектора, образующие треугольник  $\mathbf{a}=(a_1, a_2, a_3)$ ,  $\mathbf{b}=(b_1, b_2, b_3)$ ,  $\mathbf{c}=(c_1, c_2, c_3)$ . Написать программу, вычисляющую площадь проекции этого треугольника на плоскость  $Ax + By + Cz + D = 0$ .

## Задание 2.

1. Дана матрица  $\mathbf{A}(n \times n)$  и вектор  $\mathbf{a} = (a_1, a_2, \dots, a_n)$ . Написать программу вычисления вектора  $\mathbf{b} = \mathbf{A} \cdot \mathbf{a}$ .
2. Дана матрица  $\mathbf{A}(n \times n)$ . Построить  $n$ -мерный вектор по правилу: если в строке матрицы с номером  $i$  есть отрицательные элементы, то  $b_i = 0$ , в противном случае  $b_i = 1$ .
3. Даны две матрицы  $\mathbf{A}(n \times n)$  и  $\mathbf{B}(n \times n)$ . Написать программу нахождения произведения этих матриц  $\mathbf{D} = \mathbf{A} \cdot \mathbf{B}$ .
4. Дана матрица  $\mathbf{A}(n \times n)$ . Написать программу, которая меняет  $k$ -ю и  $m$ -ю строки матрицы, а затем транспонирует матрицу.
5. Дана матрица  $\mathbf{A}(n \times n)$  и вектор  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ . Написать программу нахождения скалярного произведения  $(\mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x})$ .

6. Дана матрица  $A(n \times n)$ . Написать программу нахождения матрицы  $A^T * A$ , где  $A^T$  - транспонированная матрица.
7. Даны две матрицы  $A(n \times n)$  и  $B(n \times n)$ . Написать программу нахождения матрицы, равной  $(A-B)^T$ .
8. Даны два вектора  $b = (b_1, b_2, \dots, b_n)$ ,  $x = (x_1, x_2, \dots, x_n)$  и матрица  $A(n \times n)$ . Написать программу вычисления длины вектора  $Ax - b$ .
9. Дана матрица  $A(n \times n)$ . Написать программу, которая вычеркивает столбец с номером  $p$  и переставляет остальные так, чтобы получилась матрица  $n \times (n-1)$ .
10. Даны две матрицы  $A(n \times n)$  и  $B(n \times n)$ , а также два вектора  $x = (x_1, x_2, \dots, x_n)$  и  $y = (y_1, y_2, \dots, y_n)$ . Написать программу нахождения скалярного произведения  $(Ax)^T (By)$ .
11. Даны две матрицы  $A(n \times n)$  и  $C(n \times n)$ . Написать программу вычисления матрицы  $C^T * (A+C)$ , где  $C^T$  - транспонируемая матрица.
12. Дана матрица  $A(n \times n)$ . Написать программу, которая находит максимальный по абсолютной величине элемент и переставляет строки и столбцы так, чтобы он оказался в левом верхнем углу.
13. Даны две матрицы  $A(n \times n)$  и  $B(n \times n)$ . Написать программу нахождения суммы диагональных элементов  $A * B$ .
14. Дана матрица  $A(n \times n)$ . Написать программу поиска одинаковых элементов в этой матрице.
15. Дана матрица  $A(n \times n)$ . Написать программу нахождения максимального элемента матрицы, принадлежащего отрезку  $[p, q]$ .

### Задание 3.

1. Дана матрица  $A(n \times n)$ . Написать программу нахождения минимального элемента из  $(\max_1, \dots, \max_n)$ , где  $\max_i$  - максимальный в  $i$ -той строке.
2. Даны две матрицы  $A(n \times n)$  и  $B(n \times n)$ . Написать программу получения коммутатора  $AB$  этих матриц.
3. Дана матрица  $A(n \times n)$ . Написать программу, которая упорядочивает строки этой матрицы по убыванию первых элементов строк.
4. В матрице  $A(n \times n)$  найти максимальный и минимальный элементы. Указать их разность, а также строки и столбцы, на пересечении которых они находятся.
5. Даны квадратные матрицы  $A(n \times n)$ ,  $B(n \times n)$ ,  $C(n \times n)$ . Написать программу вычисления матрицы  $(A+B)^T * C$ .
6. Дана матрица  $A(n \times n)$ . Написать программу вычисления матрицы  $AK$ , где  $K > 0$  - целое число.
7. Дана матрица  $A(n \times n)$ . Написать программу, которая упорядочивает элементы этой матрицы по возрастанию.

8. Дана матрица  $A(n \times n)$ . Написать программу, которая находит максимальный и минимальный элементы этой матрицы и переставляет столбцы и строки так, чтобы эти элементы поменялись местами.
9. Дана матрица  $A(n \times n)$ . Написать программу, которая находит максимальную сумму элементов, стоящих на диагоналях, параллельных данной.
10. Даны две матрицы  $A(n \times n)$  и  $B(n \times n)$ . Написать программу нахождения максимального значения  $\max(S_1, S_2, \dots, S_n)$ , где  $S_k$  - скалярное произведение  $k$  - й строки матрицы  $A$  на  $k$  - й столбец матрицы  $B$ .
11. Даны  $m$  векторов  $x_1 = (x_{11}, x_{21}, \dots, x_{n1})$ , ...,  $x_m = (x_{1m}, x_{2m}, \dots, x_{nm})$ . Написать программу поиска вектора минимального по длине.
12. Даны две матрицы  $A(n \times n)$  и  $B(n \times n)$ . Написать программу вычисления матрицы, где норма  $\|C\|$  матрицы  $C$  есть  $\max\|C_{km}\|$ .
13. Дана матрица  $A(n \times n)$  с положительными элементами. Написать программу, которая находит среди элементов матрицы тройки таких элементов  $(a_{ji-1}, a_{ji}, a_{ji+1})$ , чтобы существовал треугольник со сторонами, равными этим числам.
14. Даны три вектора  $a = (a_1, a_2, \dots, a_n)$ ,  $b = (b_1, b_2, \dots, b_n)$ ,  $c = (c_1, c_2, \dots, c_n)$ . Написать программу, которая проверяет зависимость этих векторов.

#### Задание 4.

1. Дана прямоугольная матрица  $A(n \times n)$ . Заменить наименьший элемент каждой строки, начиная со второй, наибольшим элементом предыдущей строки.
2. Дана действительная квадратная матрица порядка  $n$ . Построить последовательность действительных чисел  $A_1, A_2, \dots, A_n$  по правилу: если в  $i$  - той строке матрицы элемент, принадлежащий главной диагонали, отрицателен, то  $A_i$  равно сумме элементов  $i$  - той строки, предшествующих первому отрицательному элементу; в противном случае  $A_i$  равно сумме последних элементов  $i$  - той строки, начиная с первого по порядку неотрицательного элемента.
3. Дана действительная квадратная матрица порядка  $n$ . Вычислить сумму тех ее элементов, расположенных на главной диагонали и выше нее, которые превосходят по величине все элементы, расположенные ниже главной диагонали. Если на главной диагонали и выше нее нет элементов с указанным свойством, то ответом должно служить сообщение об этом.
4. Заданы квадратная матрица  $A$  порядка  $n$  и число  $K(1 \leq K \leq n)$ . Столбец с минимальным по модулю элементом в  $K$  - той строке переставить с  $K$  - тым столбцом.
5. Задана квадратная матрица порядка  $n$ . Исключить из нее строку и столбец, на пересечении которых расположен минимальный элемент главной диагонали.
6. Дана матрица  $A(n \times n)$ . Найти максимальный по модулю элемент матрицы. Переставить строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент матрицы был расположен на пересечении  $K$  - той строки и  $K$  - того столбца.

7. Дана целочисленная матрица размерностью  $n \times m$ . Найти матрицу, получающуюся из данной перестановкой строк - первой с последней, второй с предпоследней и т.д.
8. Дана целочисленная матрица размерностью  $n \times m$ , целые числа  $k, l$  ( $1 \leq k \leq n$ ,  $1 \leq l \leq m$ ,  $k < l$ ). Преобразовать матрицу так, чтобы строка с исходным номером  $k$  непосредственно следовала за строкой с исходным номером  $l$ .
9. Найти все различающиеся элементы целочисленной квадратной матрицы размерностью  $n \times m$ .
10. Дана матрица  $A(n \times n)$ . Написать программу, которая упорядочивает строки этой матрицы по убыванию первых элементов ее строк.
11. Дана матрица  $A(n \times n)$ . Написать программу, которая упорядочивает строки этой матрицы по возрастанию.
12. Дана матрица  $A(n \times n)$ . Написать программу, которая находит максимальную из сумм элементов, стоящих на диагоналях, параллельных главной диагонали.

### Функции

1. Даны действительные числа  $a, b$ . Получить  $u = \min(a, b-a)$ ,  $y = \min(ab, a+b)$ ,  $k = \min(u+v^2, 3.14)$ .
2. Даны натуральные числа  $a, b, c$ . Определить функцию  $\text{bin}(x)$ , переводящую число  $x$  из десятичной системы счисления в двоичную. Найти  $\text{bin}(a + b)$ ,  $\text{bin}(ab + c)$ .
3. Даны натуральные числа  $a, b, c$ . Найти  $\text{НОД}(a, b, c)$ , используя формулу:  
 $\text{НОД}(a, b, c) = \text{НОД}(\text{НОД}(a, b), c)$ .
4. Даны две квадратные матрицы  $A, B$  3-го порядка. Построить таблицу функции  $y = cx^2 + d$  при  $x$  меняющемся от 0 до 1 с шагом 0.1, где  $c = \text{sp}(A)$ ,  $d = \text{sp}(B)$ . ( $\text{sp}(A)$  - след матрицы  $A$  - сумма элементов главной диагонали).
5. Даны два натуральных числа  $a, b$ . Найти разность и произведение суммы цифр этих чисел. Вычисление суммы цифр числа оформить в виде функции.
6. Даны два натуральных числа  $a, b$ . Вычислить  $(a!!-ab)/(a!!+ab)$   
 Функция  $x!!$  определяется следующим образом:  
 $x!! = 1*3*5*...*x$ , если  $x$  нечетно,  
 $x!! = 2*4*6*...*x$ , если  $x$  четно.
7. Даны действительные числа  $a_0, a_1, a_2, a_3$ . Получить для  $x = 1, 3, 4$  значения  $p(x+1) - p(x)$ , где  $p(y) = a_3y^3 + a_2y^2 + a_1y + a_0$ .
8. Даны действительные числа  $a, b, c$ . Получить  $((\min(a, a+b) + \min(a, b+c)) / (1 + \min(a+bc, b)))$
9. Даны действительные числа  $a, b$ . Получить  $r = \max(a, b + a)$ ,  $d = \max(ab, a + b)$ ,  $s = \max(r + d^2, 3.14)$ .

10. Даны натуральные числа **a**, **b**, **c**. Определить функцию **bin (x)**, переводящую число **x** из десятичной системы счисления в двоичную. Найти двоичное представление этих чисел.

## Строки

### Задание 1.

1. Напишите программу подсчета суммарного числа букв '**a**' и букв '**b**' в данной строковой переменной. Вывести на экран каких букв больше.
2. Задано предложение **y**, состоящее из слов-строк. Проверить, встречается ли данное слово **x** в предложении **y**.
3. Предложение содержит буквы латинского и русского алфавитов. Написать программу, которая выводит буквы только латинского алфавита в порядке их следования в предложении.
4. Дано предложение-строка. Подсчитать количество слов, начинающихся с буквы '**a**'.
5. Написать программу, подсчитывающую, сколько раз в данном слове **x** встречается (в качестве его части) слово **y**.
6. Написать программу, которая каждый встречающийся в строке заданный символ заменяет на заданную последовательность символов, расширяя при этом строку.
7. Задано предложение-строка. Написать программу, которая находит самое длинное слово, встречающееся в предложении.
8. Написать программу, вычеркивающую из данного текста все буквы '**a**'.
9. Написать программу, которая проверяет в строке баланс открывающихся и закрывающихся круглых скобок (строка содержит арифметическое выражение).
10. Написать программу, которая каждую встреченную букву '**b**' заменяет сочетанием '**ку**'.
11. Задано предложение, состоящее из слов-строк. Написать программу, которая находит самое короткое слово в предложении.
12. Предложение состоит из слов-строк. Написать программу, которая подсчитывает количество слов в предложении.
13. Написать программу, проверяющую, является ли частью данного слова слово '**сок**'. Ответ должен быть '**да**' или '**нет**'.
14. Даны две строки. Вычеркнуть из строки **A** символы, встречающиеся в строке **B**.
15. Из данного предложения вычеркнуть слова, встречающиеся больше одного раза.

## Задание 2.

1. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст является десятичной записью числа, кратного **9**.
2. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст является записью четного числа в семеричной системе.
3. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст является записью десятичного числа, кратного **6**.
4. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст является записью десятичного числа, кратного **4**.
5. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст является шестнадцатеричной записью числа, кратного **5**.
6. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст начинается с некоторой ненулевой цифры, за которой следуют только буквы, и их количество равно числовому значению цифры.
7. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст начинается с **к** букв ( $1 \leq k \leq 9$ ), за которыми следует только одна литера-цифра с числовым значением **к**.
8. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст совпадает с начальным отрезком ряда **0123456789**(например: **0, 01, 012**).
9. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст совпадает с конечным отрезком ряда **0123456789**(например: **9, 89, 789** ).
10. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст совпадает с каким то отрезком ряда **0123456789**(например: **2, 678, 89**).
11. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст состоит только из цифр, причем их числовые значения образуют арифметическую прогрессию (например: **2468, 741, 3**).
12. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: текст содержит (помимо букв) только одну цифру, причем ее числовое значение равно длине текста.



13. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству: сумма числовых значений цифр, входящих в текст, равна длине текста.

### Записи

1. Составить программу, выводящую на экран меню детского кафе (наименование изделия, вес, стоимость).
2. Составить программу, выводящую на экран студенческую ведомость (Ф. И. О., оценки за три экзамена, средний балл).
3. Составить программу, выводящую на экран расписание движения поездов (станция отправления, станция прибытия, время прибытия, время в пути).
4. Составить программу, выводящую на экран меню ресторана "Дракон" (наименование изделия, вес, стоимость).
5. Составить программу, выводящую на экран анкетные данные учеников (Ф. И. О., год рождения, адрес, сведения о родителях).
6. Составить программу, выводящую на экран список книг домашней библиотеки (автор, название книги, издательство, год издания, стоимость).
7. Составить программу, выводящую на экран расписание экзаменов и зачетов (предмет, вид отчетности, число, преподаватель).
8. Составить программу, выводящую на экран сведения о студентах (Ф. И. О., курс, группа, номер зачетки, средний балл).
9. Составить программу, выводящую на экран сведения о периодических изданиях (наименование издания, тираж, годовая стоимость).
10. Составить программу, выводящую на экран расписание учителя (номер урока, время начала урока, класс, предмет, номер кабинета).
11. Составить программу, выводящую на экран расписание полетов самолетов (пункт посадки, время отправления, время прибытия, время полета, стоимость билета).
12. Составить программу, выводящую на экран перечень товаров, имеющих в продаже в магазине "Океан" (наименование, единица измерения, цена, количество).
13. Составить программу, выводящую на экран информацию о наличии товаров на складе (наименование, артикул, дата получения, единица измерения, количество, цена).
14. Составить программу, выводящую на экран "Телефонный справочник" (Ф. И. О., адрес, номер телефона).
15. Составить программу, выводящую на экран график отпусков (Ф. И. О., дата начала отпуска, дата выхода на работу, количество дней).

## Множества

1. Дан текст-строка из латинских букв и других знаков. Подсчитать сколько букв, сколько знаков.
2. Дан текст из цифр и строчных латинских букв. Определить, каких букв - гласных (**a, e, i, o, u**) или согласных больше в этом тексте.
3. Дана строка, состоящая из букв и цифр. Получить две новые строки, одна из которых содержит только буквы, другая только цифры.
4. Дан текст из строчных латинских букв и других знаков. Вычеркнуть из текста все гласные буквы.
5. Дан текст из букв и знаков. Подсчитать общее количество цифр и знаков '+', '-', '\*', входящих в текст.
6. Дано **30** целых чисел от **1** до **20**. Подсчитать, сколько среди них чисел, делящихся на **3**.
7. Дано **30** целых чисел от **1** до **30**. Определить, сколько среди них четных чисел.
8. Дан текст-строка, состоящая из строчных латинских букв, цифр и знаков. Получить новый текст, состоящий только из букв.
9. Дано **30** целых чисел от **1** до **50**. Определить, сколько среди них чисел, первая цифра которых **2**.
10. Дана строка, состоящая из букв и цифр, и строка из цифр и других знаков. Получить новую строку из символов первой строки, которых нет во второй.
11. Даны целые числа от **1** до **40**. Получить случайным образом **5** чисел из этого множества.
12. Даны целые числа от **8** до **22**. Получить все простые числа из этого диапазона и все остальные.
13. Дано **20** целых чисел от **1** до **40**. Определить, сколько среди них чисел, первая цифра которых **1**.
14. Дан текст-строка, состоящая из строчных латинских букв, цифр и знаков. Получить новый текст, состоящий только из цифр.
15. В возрастающем порядке напечатать все целые числа из диапазона **1..100**, представимые в виде  $n^2+m^2$  ( $n, m \geq 0$ ).

## Рекурсии

1. Даны два натуральных числа  $N > 1$  и  $M > 1$ . Написать программу нахождения **НОД(N, M)**. Использовать соотношения:
  - 1)  $\text{НОД}(N, N) = N$ ;
  - 2)  $\text{НОД}(N, M) = \text{НОД}(N-M, M)$ , если  $N > M$ ;
  - 3)  $\text{НОД}(N, M) = \text{НОД}(N, M-N)$ , если  $M > N$ ;



2. Написать функцию **f(n)**, которая рекурсивно вычисляет факториал **n**. Найти сумму **a! + b!**, где **a** и **b** - данные натуральные числа.
3. Найти **C<sub>6</sub><sup>3</sup> + C<sub>5</sub><sup>3</sup>**
4. Даны два натуральных числа **a > 1** и **b > 1**. Написать программу нахождения **НОД(a, b)**. Использовать соотношения:
  - 1) **НОД(a, a) = a**;
  - 2) **НОД(a, b) = НОД(a-b, b)**, если **a > b**;
  - 3) **НОД(a, b) = НОД(a, b-a)**, если **b > a**;
5. Написать функцию **f(n)**, которая рекурсивно вычисляет факториал **n**. Найти сумму **t! + k!**, где **t** и **k** - данные натуральные числа.
6. Найти **C<sub>8</sub><sup>7</sup> + C<sub>7</sub><sup>4</sup>**
7. \*Даны две функции **f1(x) = ln(x<sup>2</sup> - 2x + 1)**, **f2(x) = sin(cos x) - 0.7**. Найти с точностью **eps** корни уравнения **f1(x) = 0** (на отрезке **[1, 3]**) и **f2(x) = 0** (на отрезке **[0.2; 1]**), с помощью рекурсивной функции, находящей корень методом деления отрезка пополам.

## Файлы

### Задание 1 (типизированные файлы).

1. В файле **f** записаны целые числа. Написать программу, которая в файл **g** записывает четные числа, а файл **h** - нечетные.
2. Символьный файл содержит пробелы. Сжать этот файл (убрать пробелы).
3. В файле **f** записан массив действительных чисел. Найти наименьшее из этих чисел.
4. В файле **f** записаны целые числа. Написать программу, которая в файл **g** записывает положительные числа, а файл **h** - отрицательные.
5. Вычислить сумму квадратов чисел от **1** до **30**. Квадраты чисел записаны в файл.
6. Сформировать файл последовательности чисел, в котором каждая **k**-я компонента определяется по формуле: 
$$y_k = \frac{x_k^2 + 1}{x_k^3 + 2}$$
, где **x<sub>k</sub>** - **k**-я компонента исходного файла.
7. Файл **f** содержит буквы латинского и русского алфавитов. Занести в файл **h** только латинские буквы, а в файл **g** - только русские буквы.
8. Символьный файл содержит различные буквы латинского алфавита. Убрать из этого файла все буквы **'a'**.
9. Файл содержит буквы латинского алфавита. Подсчитать количество букв **'p'** в файле.
10. Файл содержит действительные числа. Найти среднее геометрическое всех положительных чисел файла.
11. Найти сумму чисел, предшествующих первому отрицательному числу в файле.

12. Найти произведение чисел из файла, меньших заданного числа **b**.
13. Файл содержит буквы латинского алфавита. Заменить буквы 'p' на 'w'.
14. Файл содержит **10** чисел, упорядоченных по возрастанию. Добавить к этим числам еще одно число, чтобы сохранилась упорядоченность по возрастанию.
15. В файле **f** элементы упорядочены по возрастанию. Требуется слить эти файлы в один файл, также упорядоченный по возрастанию.

## Задание 2 (текстовые файлы).

- Считая, что непустой текстовый файл разбит на строки, длина каждой из которых не превосходит **80**, написать процедуру, которая, дополняя короткие строки файла **f1** пробелами справа, формирует текстовый файл **f2**, все строки в котором имеют длину **80**.
- В текстовом файле записана непустая последовательность действительных чисел, разделенных пробелами (пробелов может быть больше одного). Написать функцию для нахождения наибольшего из этих чисел.
- В текстовом файле **t1** записана последовательность целых чисел, разделенных пробелами (пробелов может быть больше одного). Написать процедуру, записывающую в текстовый файл **t2** все положительные числа из **t1**.
- Написать процедуру, которая построчно вписывает содержимое непустого текстового файла **t1** в файл **t2**, вставляя в начало каждой строки ее порядковый номер (он должен занимать **4** позиции) и пробел.
- Имеется текстовый файл **T**. Написать программу, которая выводит на экран первую из самых коротких его строк.
- Дан текстовый файл **f**, содержащий программу на языке Паскаль. Проверить эту программу на соответствие открывающихся и закрывающихся круглых скобок. Считать, что каждый оператор программы может занимать произвольное число строк файла **f**.
- Дан текстовый файл **f**. Переписать в файл **g** все компоненты файла **f** с заменой в них символа '0' на символ '1' и наоборот.
- Дан текстовый файл **f**. Получить самую длинную строку файла. Если в файле имеется несколько строк с наибольшей длиной, то получить первую из них.
- Дан текстовый файл **i**. Определить количество слов в файле. Сформировать файл **g**, записав в каждой его строке по одному слову из файла **f**.
- Даны два текстовых файла **f** и **g**. Определить, совпадают ли строки файла **f** со строками файла **g**. Если нет, то получить номер первой строки и позицию первого символа в этой строке, в которых файл **f** и **g** отличаются между собой. В случае, когда один из файлов имеет **n** строк и совпадает с началом другого, более длинного, ответом должно быть **n+1**.
- Даны текстовый файл **f** и строка **s**. Получить все строки файла **f**, содержащие в качестве фрагмента строку **s**.
  - Дан текстовый файл **f1**. Исключить пробелы, стоящие в концах его строк. Результат поместить в файл **f2**.

## Динамические структуры

### Задание 1 (линейные списки).

1. Написать процедуру сортировки линейного списка по возрастанию. Элементами списка являются целые числа.
2. Построить копию данного линейного списка (с сохранением порядка следования элементов).
3. Написать процедуру, которая по двум данным линейным спискам формирует новый список, состоящий из элементов, входящих в оба данных списка одновременно. (Включать в том порядке в каком они будут найдены).
4. Написать процедуру, которая оставляет в линейном списке только первые вхождения совпадающих элементов.
5. Написать процедуру, которая по двум линейным спискам **L1**, **L2** формирует новый список, состоящий из элементов, входящих в **L1** но не входящих в **L2**.

6. Написать процедуру, которая в линейном списке из каждой группы подряд идущих равных элементов оставляет только один.
7. Написать процедуру, которая по двум линейным спискам **L1**, **L2** формирует новый список, включая в него по одному разу элементы, которые входят хотя бы в один из списков **L1**, **L2**.
8. Написать процедуру, которая удаляет из списка элементы, входящие в него только по одному разу.
9. Написать процедуру, которая вставляет в список **L** за первым вхождением данного элемента (если такое есть) все элементы списка **L1**.
10. Написать процедуру, которая формирует список **L**, включая в него по одному разу элементы, которые входят в один из списков **L1**, **L2**, но в то же время не входят в другой.
11. Написать процедуру, которая в списке **L** заменяет первое вхождение списка **L1** (если такое есть) на список **L2**.
12. Написать процедуру, которая заданный в файле текст (за ним следует точка) распечатывает в обратном порядке.

## **Задание 2 (связанные списки, двусвязные кольца).**

1. Написать программу посещения студентами дисплейного класса. Учесть приход и уход студентов в течение занятий. Данные представить в виде линейного списка.
2. Написать программу учета сдачи экзаменов студентами одной группы. Данные представить в виде очереди.
3. Написать программу учета движения троллейбуса у остановки. Данные представить в виде очереди.
4. Написать программу учета заполнения пассажирами мест в автобусе. Места в автобусе представить в виде списка. Программа учитывает освобождающиеся и занимаемые места.
5. Создается список книг, имеющихся на полке. Предусмотрите добавление книг или их изъятие. Программа учитывает наличие книг на полках.
6. Написать программу учета движения автобусов у остановки. Данные представить в виде очереди.
7. Напишите программу учета прибывших самолетов (Москва - Киев и т.д.). Самолеты становятся в очередь для заправки, заправляются и выбывают из очереди.
8. Написать программу учета продажи и поступления товаров в магазин. Перечень товаров, имеющихся в магазине, оформить в виде списка.
9. Студент путешествует по городам и посещает музеи. Пройденные города записываются в виде списка, где отмечаются музеи. Программа распечатывает в данное время список городов и музеев.

10. Написать программу учета "указаний сверху". Программа учитывает новые указания и удаляет выполненные.
11. Написать программу, учитывающую наличие студентов в читальном зале. Программа должна отражать смену занимающихся.
12. Список содержит фамилии студентов группы. Студенты получают стипендию и список соответственно корректируется. Студент, получивший стипендию, удаляется из списка.
13. Создается список из  $n$  целых чисел. Убрать из списка числа, делящиеся на  $m$ . Процедура может повторяться.
14. Создается список из фамилий. Переупорядочить список так, чтобы фамилии располагались по алфавиту.
15. Вводится  $n$  целых чисел в любом порядке и записывается в виде списка. Переупорядочить список так, чтобы числа в списке не убывали или не возрастали.