# Научно-исследовательский университет МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ

**Институт дистанционного и дополнительного образования**

## Расчетно-графическая работа по курсу

**«Программирование»**

**Тема: Разработка алгоритмов и их кодирование на алгоритмическом языке СИ**

### Выполнил: студент Петров М.Г. Принял: преподаватель Буданова Н.А.

Москва, 2019 год

**Содержание**

[Условие задачи 3](#_bookmark0)

Анализ данных Ошибка! Закладка не определена.

[Блок-схема алгоритма 4](#_bookmark1)

Раскрытие обобщенного блока Ошибка! Закладка не определена.

Форма ввода Ошибка! Закладка не определена.

Форма вывода Ошибка! Закладка не определена.

Паскаль –программа в консольном приложении Delphi без классов Ошибка! Закладка не

определена.

Паскаль-программа в консольном приложении с классом, описывающим геометрическую фигуру – квадрат Ошибка! Закладка не определена.

[Описание оконного приложения 5](#_bookmark2)

Формы оконного приложения **Ошибка! Закладка не определена.**

Автономный модуль **Ошибка! Закладка не определена.**

Описание основной формы приложения **Ошибка! Закладка не определена.**

Графический вывод **Ошибка! Закладка не определена.**

Тесты **Ошибка! Закладка не определена.**

# Раздел 1. Разработка циклических алгоритмов

## Задача 1

**Условие**

Пусть в некоторой стране А чел. страдает некоторым заболеванием. Предполагается снижение этого числа на р % за счет профилактики заболевания и на Q чел. в результате применения новых методов лечения. Определить, через сколько лет число больных уменьшится в 2 раза. Рассматривать срок не более чем 20 лет.

## Состав данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| A | начальная стоимость оборудовани | вещественный | простая переменная |
| B | начальное снижение стоимости | вещественный | простая переменная |
| р | процентное уменьшение снижения  стоимости | вещественный | простая переменная |
| N | максимальное число лет | целый | простая переменная |
| Выходные данные | | | |
| i | искомый номер года | целый | простая переменная |
| Промежуточные переменные | | | |
| At | текущая стоимость оборудования | вещественный | простая переменная |
| Bt | текущее ежегодное снижение  стоимости | вещественный | простая переменная |

## Блок-схема алгоритма

начало

Ввод A,B,p,N

At:=A; Bt:=B

i:=1

At≥A/2& iN

да

да

At<A/2

Вывод 'Срок истек'

Вывод i

конец

i:=i -1

i:=i+1

Bt:=Bt-Bt\*p/100

At:=At - Bt

**Программа на СИ (консольное приложение MS Visual Studio 2008)**

#include <iostream> #include <conio.h>

using namespace std; void main()

{

double A,B,p,At,Bt; int N,i; cout<<"A, B, p, N\n";

cin>>A>>B>>p>>N;

At=A; Bt=B; i=1;

while (At>=A/2 && i<=N)

{

At=At-Bt; Bt=Bt-Bt\*p/100; i=i+1;

}

if (At<A/2)

{

}

else

\_getch();

### }

**Задача 2**

i=i-1; cout<<"i="<<i;

cout<<"Time is over"<<endl;

## Условие

В векторе X подсчитать число нулей и заменить отрицательные элементы их абсолютными величинами**.**

## Состав данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| **с** | заданное значение | веществ. | простая переменная |
| **N** | число элементов массива | целый | простая переменная |
| **а** | заданный массив | веществ. | одномерный массив из 10  элементов |
| Выходные данные | | | |
| **р** | произведение элементов массива,  удовлетворяющих условию | веществ. | простая переменная |
| Промежуточные данные | | | |
| **i** | счетчик элементов массива | целый | простая переменная |
| **k** | количество элементов,  удовлетворяющих условию | целый | простая переменная |

**Блок-схема алгоритма**



начало

Ввод с, N,

a[i], i=0,…,N-1

p:=1; k:=0



i<N

нет

да

да

a[i]<c

k=0

да

вывод р

вывод ‘нет таких элементов’

конец

i:=i+1

k:=k+1

p:=p\*a[i]

i:=0`

**Программа на СИ (консольное приложение MS Visual Studio 2008)**

include <stdio.h> #include <conio.h> #include <math.h> void main()

{float a[10],c,p; int N,k,i; printf("Введите N, c\n"); scanf\_s("%d%f",&N,&c);

printf("Введите массив из %d элементов\n",N);

/\* Далее цикл для поэлементного ввода массива\*/ for (i=0; i<N; i++)

scanf\_s("%f", &a[i]);/\*Далее алгоритм по блок-схеме\*/ p=1; k=0;

for (i=0; i<N; i++) if (a[i]<c)

{ p=p\*a[i]; k=k+1;

}

if (k==0)

printf("Таких элементов нет\n"); else

printf(" p=%f \n",p);

\_getch();

}

## Задача 3

**Условие**

Вычислить общую сумму элементов тех строк матрицы, первый элемент которых положителен.

## Состав данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| **N** | число строк матрицы | целый | простая переменная |
| **М** | число столбцов | целый | простая переменная |
| **а** | заданная матрица | вещественный | двумерный массив размером  5\*5 |
| Выходные данные | | | |
| **i** | счетчик строк матрицы | целый | простая переменная |
| **s** | сумма элементов **i**-ой строки | вещественный | простая переменная |
| **k** | число строк с положительной  суммой элементов | целый | простая переменная |
| Промежуточные данные | | | |
| **j** | счетчик столбцов матрицы | целый | простая переменная |

**Блок-схема алгоритма**



5 i<N-1

нет

да

да

7 s>0

9

вывод i,s

11 вывод k

12 конец

i:=i+1

10

k:+k+1

8

i:=0

4



1 начало

Ввод N, M; a[i,j],

2 i=0,…,N-1, j=0,…,M-1

k:=0

3

|  |  |  |
| --- | --- | --- |
| 6 | вычисление s – суммы элементов i-й строки | |
|  | |  |

**Содержание блока 6**

s:=0

5

j M-1

да

7

j:=j+1

s:=s+a[i,j]

j:=0

**Программа на СИ (консольное приложение MS Visual Studio 2008)**

#include <stdio.h> #include <conio.h>

void main()

{float a[5][5],s; int N,M,k,i,j; printf("Введите N, M\n"); scanf\_s("%d%d",&N, &M);

printf("Введите матрицу %d \* %d\n",N,M);

/\* Далее цикл для поэлементного ввода матрицы\*/ for (i=0; i<N; i++)

for (j=0; j<M; j++)

scanf\_s("%f", &a[i][j]);/\*Далее алгоритм по блок-схеме\*/

k=0;

for (i=0; i<N; i++)

{

s=0;

for (j=0; j<M; j++)

s=s+a[i][j]; if (s>0)

{ k=k+1;

}

}

printf(" k=%d \n",k);

\_getch();

}

# Раздел 2. Функции Си

## Условие

Для матрицы A (7×3) вычислить общую сумму элементов строк, первый элемент которых положителен, а для матрицы B (6×4) — общую сумму элементов строк, в которых первый элемент больше единицы.

## Выбор назначения подпрограмм (функций)

Алгоритм вычисления минимальных значений строк матрицы одинаков (с точностью до обозначений и констант) для a и b. Поэтому вычисление минимальных значений строк произвольной матрицы следует оформить как функцию (назовем ее minmatr), а затем применить ее два раза к конкретным (фактическим) матрицам а и b. Алгоритм ввода одинаков для матриц а и b, поэтому ввод также оформим как функцию (назовем ее matrin). Аналогично вывод массива результатов будет осуществлять функция masout.

## Шаблоны выбранных функций

void masout(float a[ ],int n,char c);

void matrin(float [ ][5],int n,int m,char c);

void minmatr(float [ ][5],int n,int m,float min[ ]);

## Состав данных функции main

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| **a** | заданные матрицы | вещественный | двумерный массив **с**  **числом столбцов 5** |
| **b** |
| Выходные данные | | | |
| **mina** | минимальные значения  элементов строк матрицы **a(b)** | вещественный |  |
| **minb** |

**Блок-схема алгоритма функции minstr**



Передача n,m,a

i=0



нет

i<n

да

нет

j<m

нет

да

a[i][j]<min[i]

i=i+1

Передача min

j=j+1

min[i]=a[i][j]

j=1

min[i]=a[i][0]

**Однофайловая программа на СИ (консольное приложение Visual Studio 2008)**

#include <conio.h> #include <iostream>

using namespace std;

void masout(float a[ ],int n,char c);/\* если стоит ";", то это шаблон\*/

/\*первый размер у массивов-формальных параметров можно\*/

/\* не указывать, второй задан по максимуму\*/

void matrin(float [ ][5],int n,int m,char c);/\* c -имя матрицы\*/

/\* используется при выводе пояснительных фраз и приглаше-ний\*/

void minmatr(float [ ][5],int n,int m,float min[ ]);

/\* n и m во всех функциях передаются по значению, т. к. их\*/

/\*фактические значения - константы\*/

void main()

{float a[3][5],b[7][5], mina[3], minb[7];

matrin(a,3,5,'a'); matrin(b,7,3,'b'); minmatr(a,3,5,mina); minmatr(b,7,3,minb); masout(mina,3,'a'); masout(minb,7,'b');

\_getch();

}

void minmatr(float a[][5],int n,int m,float min[])

{int i,j;

for (i=0; i<n; i++)

{min[i]=a[i][0];/\* минимум каждой строки вычисляется \*/

/\* последовательным сравнением\*/

for (j=1;j<m; j++) if (a[i][j]<min[i])

min[i]=a[i][j];

}

}

void matrin(float a[][5],int n,int m,char c)

{int i,j;

cout<<"Input matrix "<<c<<" size of "<<n<<'\*'<<m<<endl; for (i=0;i<n;i++)

for (j=0;j<m;j++) cin>>a[i][j];

}

void masout(float a[],int n,char c)

{int i;

cout<<"Results for "<<c<<endl; for (i=0;i<n;i++)

cout<<a[i]<<" "; cout<<endl;

}