МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатики, вычислительной техники и прикладной математики

**ОТЧЕТ**

по учебной практике (эксплуатационная)

Сорокина Александра Егоровича

Курс 1 Группа ВМК-20

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность ОП «Программное обеспечение вычислительной техники и автоматизированных систем»

Руководитель практики от вуза Доцент кафедры информатики вычислительной техники и прикладной математики Семигузов Д.А.

г. Чита 2021

Оглавление

[Задача № 11в. 3](#_Toc83568040)

[Задача № 59е. 3](#_Toc83568041)

[Задача № 67д.](#_Toc83568042) 5

[Задача № 86в 5](#_Toc83568043)

[Задача № 200.](#_Toc83568044) 6

[Задача № 338в.](#_Toc83568045) 7

[Задача № 270в.](#_Toc83568046) 9

[Задача № 379д.](#_Toc83568047) 10

[Задача № 697.](#_Toc83568048) 12

[Задача № 441. 1](#_Toc83568049)4

[Задача № 692к. 1](#_Toc83568050)5

[Задача № 474г. 1](#_Toc83568051)7

[Задача № 516. 1](#_Toc83568052)8

[Задача № 900в. 1](#_Toc83568053)9

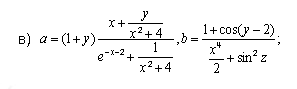
[Задача № 136к.](#_Toc83568054) 21

[Задача №](#_Toc83568055) 541г[.](#_Toc83568055) 23

[Задача № 67д#.](#_Toc83568056) 26

# Задача № 11в.

Даны x, y, z. Вычислить a, b, если



Код:

#include <stdio.h>

#include <math.h>

#include <windows.h>

// #include <iostream>

// using namespace std;

// #define M\_E 2.7182818284590452354

void main()

{

    //Просим у пользователя данные

    float x,y,z;

    printf("Это программа 11в.\nВ ней будет произведено вычисление по специальной формуле.\n");

    printf("Введите число x:");

    scanf("%f",&x);

    printf("Введите число y:");

    scanf("%f",&y);

    printf("Введите число z:");

    scanf("%f",&z);

    //Ниже произведем вычисление

    float a,b;

    a = (1+y) \* ( (x + y/(x\*x + 4)) / ( exp((x+2)\*(-1)) + (1/(x\*x + 4)) ) );

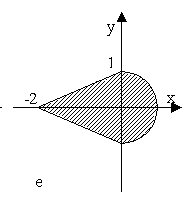
    b = (1 + cos(y-2)) / ( (pow(x, 4) / 2) + pow(sin(z), 2) );

    printf("Число а = %.2f \nЧисло b = %.2f",a , b);

}

# Задача № 59е.

Даны действительные числа x, y. Определить, принадлежит ли точка с координатами x, y заштрихованной части плоскости  
(рис.2, а - к).



Код:

#include <stdio.h>

void main()

{

    printf("Это программа 59е.\nВ ней будет определяться принадлежность точки опредленной зоне.\n");

    //Просим у пользователя данные

    float x, y;

    printf("Введите координату x:");

    scanf("%f",&x);

    printf("Введите координату y:");

    scanf("%f",&y);

    switch (x >= 0)  //Определяем координаты

    {

        case (1): //Если в правой части

        {if ( (x\*x + y\*y) <= 1)  //То смотрим по радиусу

                        printf("Точка (%.2f , %.2f) принадлежит заданной области", x, y);

                    else

                        printf("Точка (%.2f , %.2f) не принадлежит заданной области", x, y);

        } break;

        case (0):   //Если в левой то по двум прямым

        {if (y > 0)

            {        if ( y - (1/2 \*(x+2)) <= 0 )   //по верхней линии

                        {printf("Точка (%.2f , %.2f) принадлежит заданной области", x, y); break; }

                    else

                        {printf("Точка (%.2f , %.2f) не принадлежит заданной области", x, y); break; }

            }

        else

                    if ( y + (1/2 \*(x+2)) >= 0 )  //по нижней линии

                        printf("Точка (%.2f , %.2f) принадлежит заданной области", x, y);

                    else

                        printf("Точка (%.2f , %.2f) не принадлежит заданной области", x, y);

        }break;

    }

}

# Задача №67д.

Дано натуральное число n (n ≤ 100).

д) В предположении, что n ≥ 10, найти предпоследнюю цифру числа n*.*

Код:

#include <stdio.h>

#include <math.h>

void main()

{

    int N;

    printf("Введите число:");

    scanf("%d", &N);

    if ( abs(N) < 10 )

        {printf("Ну тут прикол в том что предпоследней цифры числа нет\nПонимаете? Ее просто НЕТ"); return; }

    else

        {

            N = N / 10;

            printf("Предпоследняя цифра в числе равна %d" , abs(N%10) );

        }

}

# Задача № 86в.

Дано натуральное число n.

в) Найти первую цифру числа n.

Код:

#include <stdio.h>

#include <math.h>

void main()

{

    int N;

    printf("Введите число:");

    scanf("%d", &N);

    while (abs(N) > 10)

    {

        N = N / 10;

    }

    printf("Первая цифра в числе равна %d" , N );

}

# Задача № 200.

Даны целые числа a, n, x1,...,xn(n > 0). Определить, каким по счету идёт в последовательности x1,...,xn член, равный a. Если такого члена нет, то ответом должно быть число 0.

Код:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void CreateArr(int \*Arr,int n)  //Создание массива Arr-имя массива, n-размер

{

    srand(time(NULL));      //генерируем случайные числа, чтобы они не повторялись

    for  (int i = 0 ; i < n ; i++ )

    {

        Arr[i] = rand() % 100 - 50;  //Ну и рандомим элементы массива

    }

}

void ShowArr(int \*Arr,int n)  //Вывод массива

{

   for  (int i = 0 ; i < n ; i++ )

    {

        printf("%5d",Arr[i]);

        if ((i > 0) && (i % 10 == 9))

            printf("\n");

    }

}

void Find\_Number(int \*Arr,int n, int chislo) //Находит номер искомого числа в массиве

{

    int flag = 0; //флаг для определения нашлись совпадения или нет

    for (int i = 0; i < n ; i++)

    {

        if (chislo == Arr[i])

            {

                printf("Число %d найдено, его индекс в массиве %d\n",chislo, i);

                flag++;

            }

    }

    if (flag == 0)

        printf("К сожалению мы не смогли найти совпадений. Приношу свою иииискреннее извинение");

}

void main()

{

    int n;

    printf("Введите размер массива:");  //Узнаем размер массива

    scanf("%d", &n);

    int Arr[n]; //Создаем массивчик

    CreateArr(Arr,n);

    ShowArr(Arr,n);

    int chislo;

    printf("\nВведите число которое хотите найти:");

    scanf("%d", &chislo);

    Find\_Number(Arr,n,chislo);  //Находим индекс числа

}

# Задача № 338в.

Даны натуральное число n, целые числа a1,..., a25, b1,..., bn. Среди a1,..., a25 нет повторяющихся чисел, нет их и среди b1,..., bn.

в) Получить все члены последовательности b1,..., bn, которые не входят в последовательность a1,..., a25.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

void CreateArr(int \*Arr,int n)  //Создание массива Arr-имя массива, n-размер

{

    int size = n;

    int k=0;

    while (size > 0)    //Считаем кол-во разрядов в числе размере массива

    {

        size = size / 10;

        k++;

    }

    // printf("Размеры %d  ,  %d\n", size, k);

    srand(time(NULL));      //генерируем случайные числа, чтобы они не повторялись

    int i;

    for  ( i = 0 ; i < n ; i++ )

    {

        Arr[i] = (rand() % (100\*k)) - 50;  //Ну и рандомим элементы массива

        for (int j = 0; j < i; j++ )  //проверяем есть ли повторяющиеся элементы

        {

            if (Arr[j] == Arr[i])

                {

                    i--;

                    break;

                }

                // printf("\nВошел j = %d  i = %d ", j , i);

        }

    }

    // printf("\nСоздал Размеры %d  ,  %d\n", size, k);

}

void ShowArr(int \*Arr,int n)  //Вывод массива

{

   for  (int i = 0 ; i < n ; i++ )

    {

        printf("%5d",Arr[i]);

        if ((i > 0) && (i % 10 == 9))

            printf("\n");

    }

}

void FindDifferent(int \*Arr\_1,int size\_1, int \*Arr\_2, int size\_2)   //поиск и вывод элементов массив Arr\_1, которых нет в массиве Arr\_2

{

    int flag;   //флаг для выхда из цикла при нахождении одинаковых элементов

    printf("\nМассив B без элементов массива A:\n");

    for (int i = 0; i < size\_1 ; i++)   //Идем по первому массиву

    {

        flag = 0;

        for (int j = 0; j < size\_2 ; j++)   //Ищем одинаковые во втором массиве

        {

            if (Arr\_1[i] == Arr\_2[j])

                {

                    flag++;

                    break;

                }

        }

        if (flag == 0)

            printf("%5d", Arr\_1[i]);

        if ((i > 0) && (i % 10 == 9))

            printf("\n");

    }

}

int main()

{

    setlocale(LC\_ALL,"Russian");

    int A[25];      //Создание первого массива на 25 элементов

    CreateArr(A,25);

    ShowArr(A,25);

    int n;  //Размер второго массива

    printf("\nВведите размер второго массива:");

    scanf("%d", &n);

    int B[n];

    CreateArr(B,n); //Создаем второй массив

    ShowArr(B,n);   //Выводим его на экран

    FindDifferent(B,n,A,25);    //Ищем различные элементы

    return EXIT\_SUCCESS;

}

# Задача № 270в.

Даны символы s1,s2,... Известно, что символ s1 отличен от пробела и что среди s2,s3,... имеется хотя бы один пробел. Рассматриваются s1,...,sn - символы, предшествующие первому пробелу (n заранее неизвестно). Преобразовать последовательность s1,..., sn:

в) удалив все символы, не являющиеся буквами или цифрами, и заменив каждую большую букву одноименной малой;

Код:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <locale.h>

void ChangeStr(char \*Str)   //Смена с больших букв на маленькие

{

    for(int i = 0; i < strlen(Str) ; i++)

    {

        if ((Str[i] >=65 && Str[i] <= 90) || (Str[i] >= 192 && Str[i] <= 223))

            Str[i] += 32;

    }

}

void DelStr(char \*Str)

{

    for (int i = 0 ; i < strlen(Str) ; i++)

    {

        if (!(( Str[i] >= 48 && Str[i] <= 57) || (Str[i] >= 65 && Str[i] <= 90)|| (Str[i] >= 97 && Str[i] <= 122) ))

        {

            for (int j = i; j < strlen(Str)   ; j++)

            {

                Str[j] = Str[j+1];

            }

            i--;

        }

    }

}

void main()

{

    setlocale(LC\_ALL, "Russian");

    char Str[300];

    printf("Введите строку:\n");

    gets(Str);

    //Удаление и изменение

    DelStr(Str);

    ChangeStr(Str);

    //Вывод результата

    puts("Resault");

    puts(Str);

    system("pause");

}

# Задача № 379д.

Дана действительная матрица размера n x m. Определить числа b1, ..., bm, равные соответственно:

д) разностям наибольших и наименьших значений элементов строк.

Код:

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void Create\_Matrix(int \*\*Matrix, int Row, int Col)    //Заполнение матрицы

{

    srand(time(NULL));      //генерируем случайные числа, чтобы они не повторялись

    for (int i = 0; i < Row; i++)

    {

        Matrix[i] = (int\*)malloc(Col \* sizeof(int));

        for (int j = 0; j < Col; j++)

        {

            Matrix[i][j] = rand() % 100 - 50;

        }

    }

}

void Show\_Matrix(int \*\*Matrix, int Row, int Col)    //Вывод матрицы на экран

{

    for (int i = 0; i < Row; i++)

    {

        for (int j = 0; j < Col; j++)

        {

            printf\_s("%5d", Matrix[i][j]);

        }

        printf("\n");

    }

}

int FindSub(int\*\* Matrix, int Row, int Col)

{

    int Max = -10000, Min = 10000;

    for (int j = 0; j < Col; j++)

    {

        if (Matrix[Row][j] > Max)   //Нахождение максимума

            Max = Matrix[Row][j];

        if (Matrix[Row][j] < Min)   //Нахождение Минимума

            Min = Matrix[Row][j];

    }

    return Max - Min;

}

void Show\_Resualt(int \*\*A, int Row, int Col)

{

    printf\_s("Результат:\n");

    for (int i = 0; i < Row; i++)

    {

        printf\_s("%d\n", FindSub(A, i, Col));

    }

}

void main()

{

    system("chcp 1251");

    //Узнаем размеры матрицы

    int Row, Col;

    printf\_s("Введите кол-во строк:");

    scanf\_s("%d", &Row);

    printf\_s("Введите кол-во столбцов:");

    scanf\_s("%d", &Col);

    //Создаем матрицу

    int \*\*Matrix;   //Создаем указател на матрицу

    Matrix = (int\*\*)malloc(Row \* sizeof(int\*)); //Выделение памяти под строку указателей

    Create\_Matrix(Matrix, Row, Col);

    printf("\n");

    printf("Еще раз\n");

    Show\_Matrix(Matrix, Row, Col);

    Show\_Resualt(Matrix, Row, Col);

}

# Задача №697.

Даны матрицы А и В размера *k x m*и *m x l*соответственно. Найти произведение АВ.

Код:

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void Create\_Matrix(int\*\* Matrix, int Row, int Col)    //Заполнение матрицы

{

    srand(time(NULL));      //генерируем случайные числа, чтобы они не повторялись

    for (int i = 0; i < Row; i++)

    {

        Matrix[i] = (int\*)malloc(Col \* sizeof(int));

        for (int j = 0; j < Col; j++)

        {

            Matrix[i][j] = rand() % 1000 ;

        }

    }

}

void Show\_Matrix(int\*\* Matrix, int Row, int Col)    //Вывод матрицы на экран

{

    for (int i = 0; i < Row; i++)

    {

        for (int j = 0; j < Col; j++)

        {

            printf\_s("%5d", Matrix[i][j]);

        }

        printf("\n");

    }

}

int\*\* Mult\_Matrix(int\*\* A, int RowA, int ColA, int\*\* B, int RowB, int ColB)

{

    if (ColA == RowB)

    {

        int\*\* Res;  //Создаем указатель на результирующую матрицу

        Res = (int\*\*)malloc(RowA \* sizeof(int\*));

        for (int i = 0; i < RowA; i++)

        {

            int col = 0;

            int Summ = 0;

            Res[i] = (int\*)malloc(ColB \* sizeof(int));

            for (int j = 0; j < ColB; j++)

            {

                for (int k = 0; k < ColA; k++)

                {

                    Summ += A[i][k] \* B[k][j];

                }

                Res[i][j] = Summ;

            }

        }

        return Res;

    }

    else

    {

        printf\_s("\nНевозможно перемножить матрицы!\n");

        return 0;

    }

}

void main()

{

    system("chcp 1251");

    //Узнаем размеры 1-й матрицы

    int Row\_A, Col\_A;

    printf\_s("Первая матрица:\n");

    printf\_s("Введите кол-во строк:");

    scanf\_s("%d", &Row\_A);

    printf\_s("Введите кол-во столбцов:");

    scanf\_s("%d", &Col\_A);

    //Создаем матрицу

    int\*\* Matrix\_A;   //Создаем указател на матрицу

    Matrix\_A = (int\*\*)malloc(Row\_A \* sizeof(int\*)); //Выделение памяти под строку указателей

    Create\_Matrix(Matrix\_A, Row\_A, Col\_A);

    Show\_Matrix(Matrix\_A, Row\_A, Col\_A);

    //Узнаем размеры 2-й матрицы

    int Row\_B, Col\_B;

    printf\_s("Вторая матрица:\n");

    printf\_s("Введите кол-во строк:");

    scanf\_s("%d", &Row\_B);

    printf\_s("Введите кол-во столбцов:");

    scanf\_s("%d", &Col\_B);

    //Создаем матрицу

    int\*\* Matrix\_B;   //Создаем указатель на матрицу

    Matrix\_B = (int\*\*)malloc(Row\_B \* sizeof(int\*)); //Выделение памяти под строку указателей

    Create\_Matrix(Matrix\_B, Row\_B, Col\_B);

    Show\_Matrix(Matrix\_B, Row\_B, Col\_B);

    //Перемножаем

    int\*\* Result\_Matrix;

    Result\_Matrix = Mult\_Matrix(Matrix\_A,Row\_A,Col\_A, Matrix\_B, Row\_B, Col\_B);

    printf\_s("\nРезультат умножения:\n");

    if (Result\_Matrix!= 0)

        Show\_Matrix(Result\_Matrix, Row\_A, Col\_B);

}

# Задача №441.

Дано натуральное число n. Среди чисел 1, 2, ..., n найти все те, которые можно представить в виде суммы квадратов двух натуральных чисел. (Определить процедуры, позволяющие распознавать полные квадраты).

Код:

#include <iostream>

#include <vector>

using namespace std;

void Fill\_Vector(vector<int>& Arr);        //Заполнение вектора

void Show\_Vector(vector<int> Arr);     //Вывод вектора на экран

int square(int x, int y);     //Возвращает сумму квадратов (x^2+y^2)

void ShowCheck(vector<int> Arr);      //Проверяет суммы квадратов и выводит их на экран

int main()

{

    system("chcp 1251");

    cout << "441 Дано натуральное число n. Среди чисел 1, 2, ..., n найти все те, которые можно представить в виде суммы квадратов двух натуральных чисел. (Определить процедуры, позволяющие распознавать полные квадраты)." << endl;

    int n;

    do {

        cout << "Введите число n:";

        cin >> n;

    } while (n <= 0);

    vector <int> Arr(n);        //Выделили память под n элемнтов в массиве

    Fill\_Vector(Arr);

    Show\_Vector(Arr);

    ShowCheck(Arr);

}

void ShowCheck(vector<int> Arr) {      //Проверяет суммы квадратов и выводит их на экран

    cout << "Все нужные суммы:" << endl;

    for (int i = 0; i < Arr.size(); i++) {  //Внешний цикл в котором мы идем по всему массиву

        //Два цикла которые ищут нужные числа

        for (int j = 0; j <= i; j++) {

            for (int k = 0; k <= i; k++) {

                if (i+1 == square(j+1, k+1)) {    //если нашли нужную пару чисел

                    cout << j+1 << "^2 + " << k+1 << "^2 = " << i+1 << endl;

                }

            }

        }

    }

}

void Fill\_Vector(vector<int> &Arr) {        //Заполнение вектора

    for (int i = 0; i < Arr.size(); i++) {

        Arr[i] = i+1;

    }

}

void Show\_Vector(vector<int> Arr) {     //Вывод вектора на экран

    for (int i = 0; i < Arr.size(); i++) {

        cout << Arr[i] << "   ";

    }

    cout << endl;

}

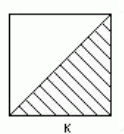
int square(int x, int y) {      //Возвращает сумму квадратов (x^2+y^2)

    return x \* x + y \* y;

}

# Задача №692к.

Дана действительная квадратная матрица порядка n. Найти наибольшее из значений элементов, расположенных в заштрихованной части матрицы (рис. 39).



Код:

#include <iostream>

#include <vector>

#include <time.h>

using namespace std;

void Fill\_Matrix(vector<vector<int>>& Matrix); //Заполнение матрицы случйными числами

void Show\_Matrix(vector<vector<int>>& Matrix); //Вывод матрицы на экран

void Find(vector<vector<int>> Mtr);    //Находит максимумы в строке и выводит их на экран

int main()

{

    system("chcp 1251");

    cout << "692 Дана действительная квадратная матрица порядка n.\n"

            "Найти наибольшее из значений элементов, расположенных в заштрихованной части матрицы(ниже побочной диагонали)\n";

    int n;

    cout << "Введите размер квадратной матрицы:";

    cin >> n;

    vector<vector<int>> Mtr(n);     //Выделяем память под матрицу

    for (int i = 0; i < Mtr.size(); i++) {

        Mtr[i].resize(n);

    }

    Fill\_Matrix(Mtr);

    Show\_Matrix(Mtr);

    Find(Mtr);

}

void Fill\_Matrix(vector<vector<int>>& Matrix) { //Заполнение матрицы случйными числами

    srand(time(nullptr));

    for (int i = 0; i < Matrix.size(); i++) {

        for (int j = 0; j < Matrix.size(); j++) {

            Matrix[i][j] = rand() % 100 - 50;

        }

    }

}

void Show\_Matrix(vector<vector<int>>& Matrix) { //Вывод матрицы на экран

    srand(time(nullptr));

    for (int i = 0; i < Matrix.size(); i++) {

        for (int j = 0; j < Matrix.size(); j++) {

            printf\_s("%6d", Matrix[i][j]);

        }

        cout << endl;

    }

}

void Find(vector<vector<int>> Mtr) {    //Находит максимумы в строке и выводит их на экран

    int n = Mtr.size();

    int Max;

    for (int i = 0; i < n; i++) {

        Max = Mtr[i][n - 1];

        for (int j = n - 1; i + j >= n - 1; j--) {

            if (Mtr[i][j] > Max) {

                Max = Mtr[i][j];

            }

        }

        cout << "Max[" << i + 1 << "] = " << Max << endl;

    }

}

# Задача 474г.

Дано натуральное n. Записать в файл g целые числа b 1 , ..., b n , определенные так, как указано в задачах а)-д) задачи [139](http://bgu-chita.ru/zadachnik/Glava06/index06.htm" \l "z139).



код:

#include <iostream>

#include <fstream>      //Библиотекчка для работы с файлами

#include <math.h>

using namespace std;

int main()

{

    system("chcp 1251");

    cout << "474.Дано натуральное n. Записать в файл g целые числа b 1 , ..., b n , 2^(i+1)\n";

    int n;

    do {

        cout << "Введите натуральное число n:";

        cin >> n;

    } while (n <= 0);

    ofstream f("Resualt.txt");  //Открываем файл для записи

    for (int i = 1; i <= n; i++) {

        f << pow(2, i + 1) << "   ";

        if (i % 10 == 0) {

            f << endl;

        }

    }

    f.close(); //Закрыли файл

}

# Задача №516.

Даны два файла f 1 и f 2. Файл f 1-это инвентарный файл, содержащий сведения о том, сколько изделий каких видов продукции хранится на складе (вид продукции задается его порядковым номером). Файл f 2-это вспомогательный файл, содержащий сведения о том, на сколько уменьшилось или увеличилось количество изделий по некоторым видам продукции. Вспомогательный файл может содержать несколько сообщений по продукции одного вида или не содержать ни одного такого сообщения. Обновить инвентарный файл на основе вспомогательного, образовав новый файл g.

Код:

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

#include <vector>

using namespace std;

int main()

{

    system("chcp 1251");

    ofstream g("g.txt");        //Итоговый файлик

    ifstream f1;                 //Для чтения из 1го файла

    ifstream f2;                 //Для чтения из 2го файла

    f1.open("f1.txt", fstream::in | fstream::out);           //открываем второй файлик

    if (!f1.is\_open()) {         //проверка на открытия файла

        cout << "Файл 1 не получилось открыть" << endl;

    }

    else {

        string str;              //строка которая будет считываться

        stringstream StrBuffer1, StrBuffer2;  //буфер для разделения строки в 1м файле и во втором

        string number1, number2;             //номер инвентаря

        int count, change;           //изменение кол-ва инвентаря

        int Summ = 0;            //Сумма изменения

        while (!f1.eof()) {      //пока не конец файла

            getline(f1, str);    //Считываем строку

            StrBuffer1 << str;  //пихаем строку в буфер, чтобы порезать ее

            StrBuffer1 >> number1 >> count; //режем строку

            cout << number1 << "    " << count << endl;

            Summ += count;      //Считаем кол-во

                f2.open("f2.txt");          //Открываем второй файл

                while (!f2.eof()) {

                    getline(f2, str);

                    StrBuffer2 << str;      //Засовываем строку в буфер, чтобы порезать

                    StrBuffer2 >> number2 >> change;

                    if (number1 == number2) {

                        Summ += change;

                    }

                    StrBuffer2.str(""); //Очистка

                    StrBuffer2.clear(); //Буфера

                }

                f2.close(); //Закрываем второй файлик

                g << number1 << " " << Summ << endl;    //Записываем результат в файл g

                Summ = 0;       //Обнулим сумму

            StrBuffer1.str(""); //Очистка

            StrBuffer1.clear(); //Буфера

            }

        f1.close();

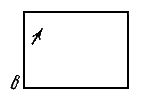
        }

}

# Задача №900в.

При работе с графическим изображением часто возникает необходимость выбрать одну или несколько точек экрана. Так, например, для того чтобы построить отрезок, следует задать два его конца, для построения окружности можно задать ее центр и любую точку на окружности и т. д. Для указывания требуемой точки обычно используют курсор. Курсор может иметь одну, из следующих конфигураций:

в) Стрелка (рис. 95, б). Указываемая точка-это точка, в которую помещается острие стрелки.



Код:

#include <iostream>

#include <Windows.h>

#include <time.h>

int main()

{

    POINT op;

    HWND hwnd = GetConsoleWindow();

    HDC pero = GetDC(hwnd);

    SelectObject(pero, GetStockObject(WHITE\_PEN));

    MoveToEx(pero, 100, 100, &op);

    LineTo(pero, 100, 175);

    for (int i = 25; i <= 100; i++) {

        MoveToEx(pero, 100, 175, &op);

        LineTo(pero, i, 100);

    }

    MoveToEx(pero, 100, 100, &op);

    LineTo(pero, 0, 200);

    //по y

    for (int i = 185; i <= 200; i++) {

        MoveToEx(pero, 85, 100, &op);

        LineTo(pero, 0, i);

    }

    //по x

    for (int i = 0; i <= 15; i++) {

        MoveToEx(pero, 100, 115, &op);

        LineTo(pero, i, 200);

    }

    //по y

    for (int i = 185; i <= 200; i++) {

        MoveToEx(pero, 100, 100, &op);

        LineTo(pero, 0, i);

    }

    //по x

    for (int i = 0; i <= 15; i++) {

        MoveToEx(pero, 100, 100, &op);

        LineTo(pero, i, 200);

    }

    //прикольный логотип

    MoveToEx(pero, 100, 100, &op);

    LineTo(pero, 100, 175);

    for (int i = 25; i <= 100; i++) {

        MoveToEx(pero, 100, 175, &op);

        LineTo(pero, i, 100);

    }

    MoveToEx(pero, 100, 100, &op);

    LineTo(pero, 0, 200);

    //по y

    for (int i = 185; i <= 200; i++) {

        MoveToEx(pero, 100, i, &op);

        LineTo(pero, 0, i);

    }

    //по x

    for (int i = 0; i <= 15; i++) {

        MoveToEx(pero, i, 100, &op);

        LineTo(pero, i, 200);

    }//\*/

    ReleaseDC(hwnd, pero);

    std::cin.get();

}

# Задача № 136к.

Даны натуральное число n, действительные числа a1,..., an. Вычислить:



Код:

#include <iostream>

using namespace std;

typedef struct  UList{

    int x;

    struct UList\* next;

}Uzel;

void add(Uzel\* Head, Uzel\* a);   //Добавление узла

void Sp\_Show(Uzel\* Head);        //Вывод списка на экран

int Sp\_Summ(Uzel\* Head);         //Вернет сумму всех полей в списков

void Sp\_Delete(Uzel\* Head);     //Удаление списка

int main()

{

    system("chcp 1251");

    Uzel\* Head = new Uzel;

    Head->x = 1;

    Head->next = NULL;

    Uzel\* current;

    int n;

    cout << "Введите n:";

    cin >> n;

    for (int i = 2; i <= n; i++) {

        current = new Uzel;

        current->x = i;

        add(Head, current);

    }

    Sp\_Show(Head);

    cout << "\nСумма = " << Sp\_Summ(Head);

    Sp\_Delete(Head);

}

void Sp\_Delete(Uzel\* Head) {    //Удаление списка

    Uzel\* current = Head;

    Head = Head->next;

    while (Head != nullptr) {

        free(current);

        current = Head;

        Head = Head->next;

    }

    free(Head);

    cout << "\nВроде бы удалил\n";

}

void add(Uzel\* Head, Uzel\* a) {       //Добавление узла в конец списка

    Uzel\* current = Head;

   if (Head == nullptr) {

        cout << "Ты ебобо? Ты голову где потерял?\n";

    }

    else

        while (current->next != nullptr) {  //Доходим до конца списка

            current = current->next;

        }

        current->next = a;  //и связываем список с последним эелементом

        a->next = nullptr;

}

void Sp\_Show(Uzel\* Head) {      //Вывод списка на экран

    Uzel\* current = Head;

    if (Head == nullptr) {

        cout << "Списка то нет!\n";

    }

    else

        while (current != nullptr) {

            cout << current->x << endl;

            current = current->next;

        }

}

int Sp\_Summ(Uzel\* Head) {   //Вернет сумму всех полей в списков

    Uzel\* current = Head;   //Текущий

    int Summ = 0;

    while (current != nullptr) {

        Summ += current->x;

        current = current->next;

    }

    return Summ;

}

# Задача №541г.

Даны натуральное число n, действительные числа a1, ..., an. Преобразовать последовательность a1, ..., an, расположив вначале отрицательные члены, а затем - неотрицательные. При этом:

г) порядок тех и других чисел изменяется на обратный.

Код:

#include <iostream>

using namespace std;

typedef struct  UList {

    int x;

    struct UList\* next;

}Uzel;

void NewUzel(Uzel\* Head);               //Выделение памяти под узел и вставка в список

void InEnd(Uzel\* Head, Uzel\* in);       //Вставка в конец списка

void Show(Uzel\* Head);                  //Вывод на экран

Uzel\* Change(Uzel\* Head);                //Изменение списка

int main()

{

    system("chcp 1251");

    Uzel\* Head = new Uzel;

    Head->x = rand() % 100 - 50;

    Head->next = nullptr;

    int n;

    cout << "Введите n:";

    cin >> n;

    for (int i = 1; i < n; i++) {

        NewUzel(Head);

    }

    Show(Head);

    Head = Change(Head);

    Show(Head);

}

void NewUzel(Uzel\* Head) {              //Выделение памяти под узел и вставка в список

    Uzel\* current = new Uzel;           //Выделили память

    current->x = rand() % 100 - 50;     //Заполнили хранилище

    current->next = nullptr;            //Обнуляем указатель

    InEnd(Head, current);               //Вставляем в конец списка

}

void InEnd(Uzel\* Head, Uzel\* in) {      //Вставка в конец списка

    Uzel\* current = Head;

    while (current->next != nullptr) {

        current = current->next;

    }

    current->next = in;

    in->next = nullptr;

}

void Show(Uzel\* Head) {                  //Вывод на экран

    Uzel\* current = Head;

    cout << "Список:\n";

    while (current != nullptr) {

        cout << current->x << endl;

        current = current->next;

    }

}

Uzel \* Change(Uzel \*Head) {               //Изменение списка

    Uzel\* current = Head;

    Uzel\* buffer = Head;

    int step = 0;

    cout << "head:\n";

    while (current != nullptr) {

        if ( (current->x >= 0) && (current != Head) ) {

            while (buffer->next != current) {               //Находим элемент перед current

                buffer = buffer->next;

            }

            buffer->next = current->next;                   //Связывае с тем что стоит за current

            //Теперь вставим current в голову списка

            current->next = Head;

            Head = current;                                 //Новая голова

            buffer = Head;

            Show(Head);

            int k = 0;

            for (int i = 0; i <= step; i++) {                //Прогоняем

                current = current->next;

                k++;

            }

            step = k;

        }

        else {

            step++;                                         //переходим к следующему

            current = current->next;

        }

    }

    cout << "После смены:\n";

    Show(Head);

    return Head;

}

# Задача №67д#.

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Program\_67d

{

    public partial class MainForm : Form

    {

        public MainForm()

        {

            InitializeComponent();

        }

        private void label1\_Click(object sender, EventArgs e)

        {

        }

        private void button\_run\_Click(object sender, EventArgs e)

        {

            string str;

            str = Edit.Text;

            int size = str.Length;

            listBox1.Items.Add(str[size - 2]);

        }

    }

}

