

11 Даны x, y, z . Вычислить a, b , если:

$$б) a = \frac{3+e^{y-1}}{1+x^2|y-\tan(z)|}, b = 1+|y-x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3};$$

```
#include <conio.h>
#include <math.h>
#include <stdio.h>
int main()
{
    float x,y,z,a,b;
    printf("Введите x,y,z= ");
    scanf("%f%f%f",&x,&y,&z);
    a=(3+exp(y-1))/((1+x)*abs(y-tan(z)));
    b=1+abs(y-x)+sqr(y-x)/2+pow(y-x,3)/3;
    printf("a=%f b=%f",a,b);
}
```

60 Пусть D - заштрихованная часть плоскости (рис.3, а - е) и пусть U определяется по x и y следующим образом (запись $(x, y) \in D$ означает, что точка с координатами x, y принадлежит D):

$$б) U = \begin{cases} -3, & \text{если } (x, y) \in D, \\ y^2 & \text{в противном случае;} \end{cases}$$

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

int main(){
    float u, x, y;
    printf("Введите x,y: ");
    scanf("%f%f",&x,&y);
    if (y<=x/2) and (sqr(y)>=1-sqr(x)){
        u=-3;
    }
    else{
        u=sqr(y);
    }
    printf("u= ",u);
}
```

77 Дано натуральное число n . Вычислить:

$$а) \sqrt{2+\sqrt{2+\dots+\sqrt{2}}} - n \text{ корней.}$$

```

#include <math.h>
#include <iostream.h>

int main(){
    double res;
    int n;
    printf("n= ");
    scanf("d",&n);
    res=0;
    for (int i=0; i<n; i++){
        res=sqrt(res+2);
    }
    printf("res= ", res);
}

```

137 Даны натуральное число n , действительные числа a_1, \dots, a_n . Вычислить: $a_1, -a_1a_2, a_1a_2a_3, \dots, (-1)^{n+1}a_1a_2\dots a_n$;

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

typedef struct Node
{
    int x;
    struct Node *next;
};
typedef struct Node* PNode;

PNode createList(int n) {
    PNode a, p, head;
    head = NULL; p = NULL;
    int x,i;
    for (i = 1; i <= n; i++) {
        a = (PNode)calloc(1, sizeof(PNode));
        if (head == NULL) {
            head = a;
            p = head;}
        else {
            p->next = a;
            p = p->next;}
        a->next = NULL;
        printf(" i);
        scanf("%d", &x);
        a->x = x;
    }
    return head;
}

```

```

PNode calculate(PNode head, int n) {
    PNode headB, p, pb, b;
    int i, j, result;
    headB = NULL;
    pb = NULL;
    p = head;
    result = 1;
    for (i = 1; i <= n; i++) {
        b = (PNode)calloc(1, sizeof(PNode));
        if (headB == NULL) {
            headB = b;
            pb = headB;
        }
        else {
            pb->next = b;
            pb = pb->next;
        }
        b->next = NULL;
        result = result * p->x;
        p = p->next;
        pb->x = pow(-1, i + 1) * result;
    }
    return headB;
}

void printList(PNode head) {
    PNode p;
    p = head;
    while (p != NULL) {
        printf("%d ", p->x);
        p = p->next;
    }
    printf("\n");
}

int main() {
    system("chcp 1251");
    int n;
    PNode head, headB;
    head = NULL;
    scanf("%d", &n);
    head = createList(n);
    headB = calculate(head, n);
    printList(head);
    printList(headB);
    return 0;
}

```

197 Вычислить $\sum_{i=1}^{30} (a_i - b_i)^2$, где

$$a_i = \begin{cases} i, & \text{если } i - \text{нечётное} \\ i/2 & \text{в противном случае} \end{cases}$$

$$b_i = \begin{cases} i^2, & \text{если } i - \text{нечётное} \\ i^3 & \text{... в противном случае} \end{cases}$$

```
#include <stdio.h>
```

```

int main()
{
    float res, at, bt;
    int a[30];
    int b[30];
    res = 0;
    for (int i = 1; i <= 30; i++){

```

```

        at = 0;
        bt = 0;
        if (a[i] % 2 == 1) at = a[i];
        else at = a[i] / 2;
        if (b[i] % 2 == 1) bt = b[i] * b[i];
        else bt = b[i] * b[i] * b[i];
        res = res + pow(at - bt, 2);
    }
    printf("%f", res);
    return 0;
}

```

264 Даны натуральное число n , символы s_1, \dots, s_n . Исключить из последовательности s_1, \dots, s_n группы символов, группы символов, расположенные между скобками $(,)$. Сами скобки тоже должны быть исключены. Предполагается, что внутри каждой пары скобок нет других скобок.

```

#include <stdio.h>
#include <stdbool.h>

int main(void)
{
    const char* s = "ab(ra)ca(dab)ra";
    bool inside_brackets = false;
    for (char* i = s; *i != '\0'; i++)
    {
        switch (*i)
        {
            case '(': inside_brackets = true; break;
            case ')': inside_brackets = false; break;
            default: if (!inside_brackets) { putchar(*i); } break;
        }
    }
    putchar('\n');
    return 0;
}

```

334 ВЫЧИСЛИТЬ:

$$0) \sum_{i=1}^{100} \sum_{j=1}^{60} \sin(i^3 + j^4).$$

```

#include <math.h>
#include <stdio.h>

```

```

int main(){
    float res=0;
    int n, m;
    printf('Введите n и m: ');
    scanf('%d%d', &n, &m);
    for (int i=0; i<=n ;i++){
        for (int j=0; j<=m ;j++){
            res=res+sin((pow(i,3)+pow(j,4)));
        }
    }
    printf('Res: ', res);
}

```


393 Даны натуральное число n , целочисленная квадратная матрица порядка n . Получить b_1, \dots, b_n , где b_i – это значение первого по порядку положительного элемента i -й строки (если таких элементов нет, то принять $b_i = 1$);

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char* argv[]) {
    const int n = 5;
    int A[n][n];
    const int r_limit = 100;
    int i, j;
    srand((unsigned) time(NULL));
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            A[i][j] = rand()%r_limit - rand()%r_limit;
        }
    }
    printf("Initial matrix:\n");
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++)
            printf("%d\t", A[i][j]);
        printf("\n");
    }
    int B[n];
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++)
            if(A[i][j] > 0)
                break;
        B[i] = (j!=n) ? A[i][j] : 1;
    }
    printf("\nVector B:\n");
    for(i=0; i<n; i++){
        printf("%d ", B[i]);
    }
    return 0;
}
```

434 Даны действительные числа $a_1, \dots, a_n, b_1, \dots, b_m$. В последовательности a_1, \dots, a_n и в последовательности b_1, \dots, b_m все члены, следующие за членом с наибольшим значением (за первым по порядку, если их несколько) , заменить на 0.5.

```
#include <stdio.h>
#include <time.h>
#include <iostream>
using namespace std;
const int N=20;

void calc(float a[N], float max1, int k, int n){
    max1=a[0];
    for (int i=0;i<n;i++)
        if (a[i]>max1)
        {
            max1=a[i];
            k=i;
        }
    for (int i=k+1;i<n;i++)
```

```

        a[i]=0.5;
    }

int main()
{
    setlocale (LC_ALL, "Russian");
    srand(time(NULL));
    int i, j, n, m, k=0;
    float a[N], max1;

    cout<<"Введите n: ";
    cin>>n;

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

    cout<<endl;

    calc(a, max1, k, n);

    cout<<endl;
    for (i=0;i<n;i++)
        cout<<a[i]<<" ";
    cout<<endl;
}

```

473 Дан файл f, компоненты которого являются целыми числами. Найти количество четных чисел среди компонент;

```

#include <string>
#include <math.h>
#include <stdio.h>
#include <iostream>
#include <locale>
#include <fstream>
using namespace std;

#pragma warning(disable: 4996)

int main()
{
    setlocale(0, "");

    ofstream fout("f.txt");
    for (int i = 0; i < 15; i++)
    {
        fout << rand() % 100 << " ";
    }
    fout.close();
    ifstream fin("f.txt");
    int Arr[15];
    int count = 0;
    for (int i = 0; i < 15; i++)
    {
        fin >> Arr[i];
        if (Arr[i] % 2 == 0)
        {
            count++;
        }
    }
    cout << count << endl;
}

```

```

    system("pause");
    return 0;
}

```

687 Дано действительное число x . Получить квадратную матрицу порядка 10:

$$\begin{vmatrix}
 1 & x & \dots & x^8 & x^9 \\
 x & 0 & \dots & 0 & x^8 \\
 & & \ddots & & \\
 & & & \ddots & \\
 x^8 & 0 & \dots & 0 & x \\
 x^9 & x^8 & \dots & x & 1
 \end{vmatrix}$$

(Середина заполняется нулями).

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

float** createMatrix(int n)
{
    float** a;

    a = (float**)calloc(n, sizeof(float*));
    for (unsigned i = 0; i < n; i++)
    {
        a[i] = (float*)calloc(n, sizeof(float));
    }

    return a;
}

void calculateMatrix(float** a, int n, float x)
{
    float ch = 1;
    float chl;
    int k = n - 1;
    for (unsigned i = 0; i < n; i++)
    {
        for (unsigned j = 0; j < n; j++)
        {
            a[i][j] = 0;
        }
        ch = ch * x;
    }
    chl = ch / x;
    for (unsigned i = 0; i < n; i++)
    {
        a[0][i] = chl;
        a[i][0] = a[0][i];
        a[k][i] = ch;
        a[i][k] = a[k][i];
        ch = ch / x;
    }
}

```



```

        chl = chl * x;
    }
}

void printMatr(float** matr, int n)
{
    for (unsigned i = 0; i < n; i++)
    {
        for (unsigned j = 0; j < n; j++)
        {
            printf("%10.2f", matr[i][j]);
        }
        printf("\n");
    }
}

int main()
{
    system("chcp 1251");

    float** a;
    int n;
    float x;

    printf("??????? n: ");
    scanf("%d", &n);
    printf("??????? x: ");
    scanf("%f", &x);
    a = createMatrix(n);
    calculateMatrix(a, n, x);
    printMatr(a, n);

    return 0;
}

```

705 Даны квадратные матрицы A и B порядка n. Получить матрицу A(B-E)+C, где E единичная матрица порядка n, а элементы матрицы C вычисляются по формуле

$$C_{ij} = \frac{1}{i+j}, i, j = 1, 2, \dots, n.$$

```

#include <iostream>
#define MIN_SIZE 1

double** addition(double** A, double** B, int size) {
    double** result = new double* [size];

    for (int i = 0; i < size; i++) {
        result[i] = new double [size];
    }
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            result[i][j] = A[i][j] + B[i][j];
        }
    }
    return result;
}

```

```

double** multiplicationByNumber(double** A, double number, int size){
    double** result= new double* [size];

    for (int i = 0; i < size; i++) {
        result[i]=new double [size];
    }
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            result[i][j]=A[i][j]*number;
        }
    }
    return result;
}

```

```

double** multiplication(double** A, double** B, int size){
    double** result= new double* [size];
    double sum;
    for (int i = 0; i < size; i++) {
        result[i]=new double [size];
    }
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            sum = 0;
            for (int k = 0; k < size; k++) {
                sum+=A[i][k]*B[k][j];
            }
            result[i][j] = sum;
        }
    }
    return result;
}

```

```

int main(){
    double** A;
    double** B;
    double** C;
    double** E;
    double** R;

    int n;
    cout<< "Input matrix size ";
    cin>> n;
    if (n > MIN_SIZE) {
        cout<< "Initialize A" <<endl;          // ввод матрицы A
        A= new double* [n];
        for (int i = 0; i < n; i++) {
            A[i]=new double [n];
        }
        for (int i = 0; i < n; i++) {
            cout << "Input " << i << " row"<< endl;
            for (int j = 0; j < n; j++) {
                cin>>A[i][j];
            }
        }

        cout << "Matrix A " << endl<<endl;      // вывод матрицы A
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                cout<<A[i][j]<<" ";
            }
            cout<<endl;
        }
    }
}

```

```

cout<< "Initialize B" <<endl;          // ввод матрицы B
B = new double* [n];
    for (int i = 0; i < n; i++) {
        B[i]=new double [n];
    }
for (int i = 0; i < n; i++) {
    cout << "Input " << i << " row"<< endl;
    for (int j = 0; j < n; j++) {
        cin>>B[i][j];
    }
}

cout << "Matrix B" << endl<<endl;      // вывод матрицы B

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        cout<<B[i][j]<<" ";
    }
    cout<<endl;
}

cout<< "Initialize C" <<endl;          // ввод матрицы C
C = new double* [n];
    for (int i = 0; i < n; i++) {
        C[i]=new double [n];
    }
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        C[i][j]=1./(i+1)+(j+1);
    }
}

cout << "Matrix C " << endl<<endl;      // вывод матрицы C

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        cout<<C[i][j]<<" ";
    }
    cout<<endl;
}

E= new double* [n];
    for (int i = 0; i < n; i++) {
        E[i]=new double [n];
    }
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (i==j){
            E[i][j]=1;
        }else{
            E[i][j]=0;
        }
    }
}

// Вычисление результата
R = multiplication(A,addition(B,multiplicationByNumber(E,-1,n),n),n);
R = addition(R,C,n);

//Вывод результата
cout << "Matrix R " << endl<<endl;

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        cout<<R[i][j]<<" ";
    }
    cout<<endl;
}

```

```

    }
} else {
    cout<<"Incorrect input data"<<endl;
}
system("pause");
}

```

846 Исследовать область определения и построить графики следующих функций:

$$2) y = 3 - \frac{2}{x} - \frac{3}{x^2};$$

```

#include<stdio.h>
#include<stdlib.h>
#include<locale.h>
#include<windows.h>

int x = 300 , y = 150;

void MakeSM(POINT p, HDC hdc)
{
    for (int i = 0; i != 600; i += 15)
    {
        MoveToEx(hdc, i, 145, &p);
        LineTo(hdc, i, 155);
    }

    for (int i = 0; i != 300; i += 15)
    {
        MoveToEx(hdc, 295, i, &p);
        LineTo(hdc, 305, i);
    }
}

void MakeXOY(POINT p, HDC hdc)
{
    MoveToEx(hdc, x, 0, &p);
    LineTo(hdc, x, x);
    MoveToEx(hdc, 0, y, &p);
    LineTo(hdc, 2 * x, y);
}

float function(float x)
{
    return 1 * (3 - (2/x) - (3/(x*x)));
}

main()
{
    setlocale(0, "");

    HWND hwnd = GetConsoleWindow();
    HDC hdc = GetDC(hwnd);
    SelectObject(hdc, GetStockObject(WHITE_PEN));

    POINT p;

    MoveToEx(hdc, 150, 150, &p);

```

}

11 Даны x, y, z . Вычислить a, b , если

```
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 _11
{
    public partial class Form1 : Form
```

```

{
    public Form1()
    {
        InitializeComponent();
    }

    private void Label1_Click(object sender, EventArgs e)
    {
    }

    private void Button1_Click(object sender, EventArgs e)
    {
        if (textX.Text.Length == 0 || textY.Text.Length == 0 || textZ.Text.Length ==
0)
        {
            MessageBox.Show(
                "Заполните все поля",
                "Ошибка",
                MessageBoxButtons.OK,
                MessageBoxIcon.Error,
                MessageBoxDefaultButton.Button1,
                MessageBoxOptions.DefaultDesktopOnly);
        }
        else
        {
            double a, b, x, y, z;
            x = Convert.ToDouble(textX.Text);
            y = Convert.ToDouble(textY.Text);
            z = Convert.ToDouble(textZ.Text);
            a = (3 + Math.Pow(Math.E, y - 1)) / (1 + (x * x) * Math.Abs(y -
Math.Tan(z)));
            b = (1 + Math.Abs(y - x) + Math.Pow(y - x, 2) / 2 + Math.Pow(Math.Abs(y -
x), 3) / 3);

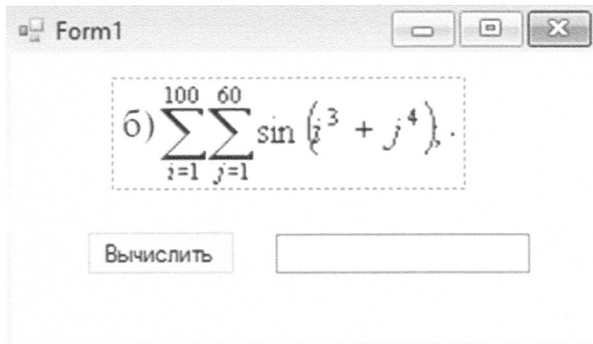
            textResultA.Text = Convert.ToString(a);
            textResultB.Text = Convert.ToString(b);
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {
    }
}

```

334 Вычислить:

$$0) \sum_{i=1}^{100} \sum_{j=1}^{60} \sin(i^3 + j^4).$$



```
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 _334
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            double res = 0;
            for (int i = 0; i <= 100; i++)
            {
                for (int j = 0; j <= 60; j++)
                {
                    res = res + Math.Sin(Math.Pow(i, 3) + Math.Pow(j, 4));
                }
            }
            textBox1.Text = Convert.ToString(res);
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

705 Даны квадратные матрицы A и B порядка n. Получить матрицу A(B-E)+C, где E единичная матрица порядка n, а элементы матрицы C вычисляются по формуле

$$C_{ij} = \frac{1}{i+j}, \quad i, j = 1, 2, \dots, n.$$

```
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 _705
{
    public partial class Form1 : Form
    {
        int n;
        double[,] a = new double[10, 10];
        double[,] b = new double[10, 10];
        double[,] e = new double[10, 10];
        double[,] c = new double[10, 10];
        double[,] res = new double[10, 10];

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            n = Convert.ToInt32(textBox1.Text);
            if (n > 10 || n < 1)
            {
                btnCalc.Enabled = false;
                MessageBox.Show(
                    "Размерность не должна превышать 10 и быть меньше 1",
                    "Ошибка",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Error,
                    MessageBoxDefaultButton.Button1,
                    MessageBoxOptions.DefaultDesktopOnly);
            }
        }
    }
}
```



```

else
{
    btnCalc.Enabled = true;

    dataGridA.RowCount = n;
    dataGridA.ColumnCount = n;
    dataGridB.RowCount = n;
    dataGridB.ColumnCount = n;
    dataGridRes.RowCount = n;
    dataGridRes.ColumnCount = n;
}
}

double[,] mulMatr(double[,] a, double[,] b)
{
    double[,] result = new double[10, 10];
    double sum;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            sum = 0;
            for (int k = 0; k < n; k++)
            {
                sum += a[i, k] * b[k, j];
            }
            result[i, j] = sum;
        }
    }

    return result;
}

double[,] subMatr(double[,] a, double[,] b)
{
    double[,] result = new double[10, 10];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            result[i, j] = a[i, j] - b[i, j];
        }
    }

    return result;
}

double[,] addMatr(double[,] a, double[,] b)
{
    double[,] result = new double[10, 10];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            result[i, j] = a[i, j] + b[i, j];
        }
    }

    return result;
}

void calcRes()

```

```

    {
        res = addMatr(mulMatr(a, subMatr(b, e)), c);

        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                dataGridRes[j, i].Value = res[i, j];
            }
        }
    }

    void calcE()
    {
        for(int i = 0; i < n; i++)
        {
            e[i, i] = 1.0;
        }
    }

    void calcC()
    {
        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                c[i, j] = 1.0 / (i + 1) + (j + 1);
            }
        }
    }

    private void btnCalc_Click(object sender, EventArgs e)
    {
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                a[i, j] = Convert.ToDouble(dataGridA[j, i].Value);
                b[i, j] = Convert.ToDouble(dataGridB[j, i].Value);
            }
        }
        calcE();
        calcC();

        calcRes();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
    }

    private void dataGridB_CellContentClick(object sender, DataGridViewCellEventArgs
e)
    {
    }
}

```