

Лабораторная работа №7

1) Тема: Функции. Введение в структуры.

8.1)

2) Вычислить минимальное значение (одно) из положительных элементов каждого из двух массивов чисел M[11] и N[11], в которых могут присутствовать отрицательные числа.

3)

```
int Small(int num[11])
```

```
{
```

```
    int a=num[0];
```

```
    for (int i=0; i<11;i++)
```

```
    {
```

```
        if (num[i]>=0)
```

```
        {
```

```
            if(num[i]<a)
```

```
            {
```

```
                a=num[i];
```

```
            }
```

```
        }
```

```
    }
```

```
    return a;
```

```
}
```

4)

Имя	Смысл	Тип
Small	Функция, вычисляющая минимальный положительный элемент массива	int Small(int num[11])
num[11]	Массив, используемый в функции	Массив элементов типа int
a	Значение, выдаваемое функцией	Int
i	Индекс массивов	Int
sM	Минимальное число в M	Int

sN	Минимальное число в N	Int
M[11]	Массив элементов M	Массив элементов типа int
N[11]	Массив элементов N	Массив элементов типа int

5)

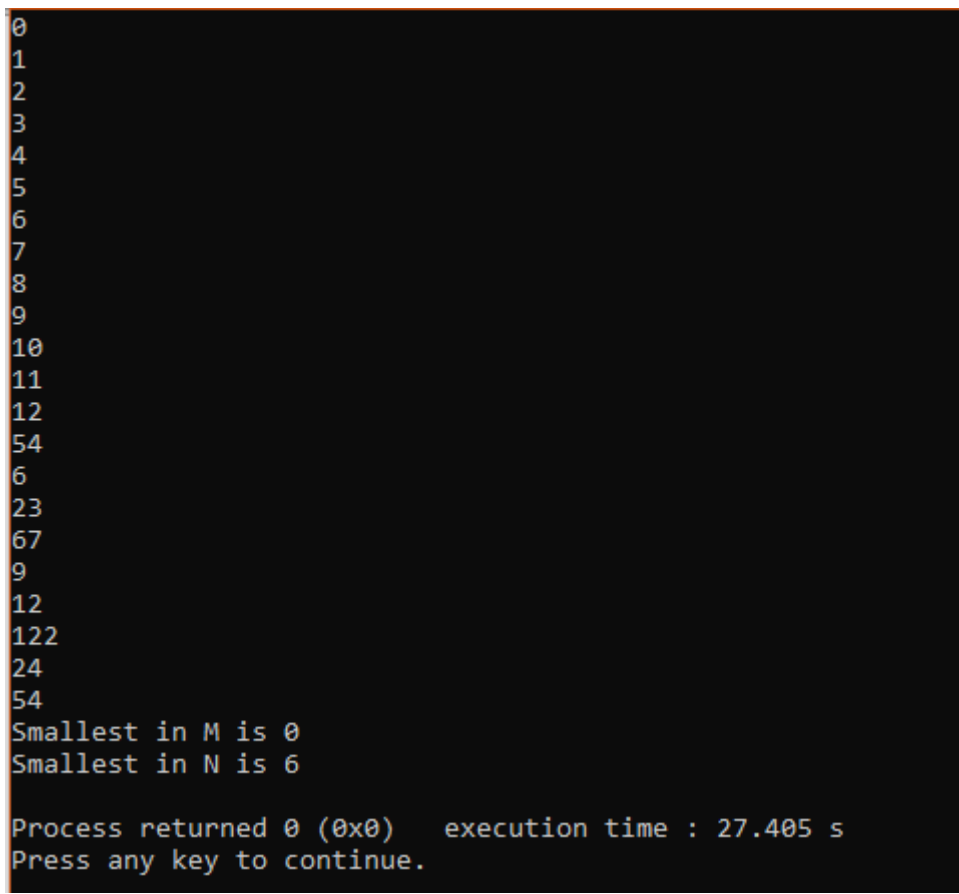
```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int Small(int num[11])
{
    int a=num[0];
    for (int i=0; i<11;i++)
    {
        if (num[i]>=0)
        {
            if(num[i]<a)
            {
                a=num[i];
            }
        }
    }
    return a;
}
main()
{
    int a,i,sM,sN;
    int M[11];
    int N[11];
    for (i=0; i<11;i++)
    {
```

```

    scanf("%d", &M[i]);
}
for (i=0; i<11;i++)
{
    scanf("%d", &N[i]);
}
sM=Small(M);
printf("Smallest in M is %d\n",sM);
sN=Small(N);
printf("Smallest in N is %d\n",sN);
return 0;
}

```

6)



```

0
1
2
3
4
5
6
7
8
9
10
11
12
54
6
23
67
9
12
122
24
54
Smallest in M is 0
Smallest in N is 6

Process returned 0 (0x0)   execution time : 27.405 s
Press any key to continue.

```

8.2)

2) Вычислить

$$D = P^3(a[i][j], b[i][j]) - P(i, j),$$

где $i = [0; 4]$, $j = [0; 5]$, если

$$P(k, h) = h \frac{1}{\sqrt{k}} - 1.$$

3)

$$D = P^3(a[i][j], b[i][j]) - P(i, j),$$

где $i = [0; 4]$, $j = [0; 5]$, если

$$P(k, h) = h \frac{1}{\sqrt{k}} - 1.$$

4)

Имя	Смысл	Тип
P	Функция, вычисляющая значение P	double P(int x, int y)
x	Переменная функции	Int
y	Переменная функции	Int
z	Результат функции	Int
i	Индекс элемента массива	Int
j	Индекс элемента массива	Int
A	Массив чисел a	Массив чисел типа int
B	Массив чисел b	Массив чисел типа int
S	Значение выражения	Double
A[i][j]	Элемент Массива a	Int
B[i][j]	Элемент Массива b	Int

5)

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

```
#include <stdlib.h>
```

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

```
double P(int x, int y)
```

```
{
```

```
    double z;
```

```
    z=y*(1/pow(x,1/2))-1;
```

```
    return(z);
```

```
}
```

```
main()
```

```
{
```

```
    int i,j;
```

```

int A[5][6];
int B[5][6];
double S;
printf("Enter matrix A \n");
for(i=0;i<5;i++)
{
    for (j=0;j<6;j++)
    {
        scanf("%d",&A[i][j]);
    }
}
printf("Enter matrix B \n");
for(i=0;i<5;i++)
{
    for (j=0;j<6;j++)
    {
        scanf("%d",&B[i][j]);
    }
}
for (i=0;i<5;i++)
{
    for (j=0;j<6;j++)
    {
        S=pow(P(A[i][j],B[i][j]),3)-P(i,j);
        printf("\n P[%d][%d]=%f",i,j,S);
    }
}
return 0;
}

```

6)

Enter matrix A

1
2
3
4
5
6
7
8
9
10
9
8
7
6
5
4
3
2
1
2
3
4
5
6
7
8
9
10
9
8

Enter matrix B

```
Enter matrix B
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
2
27
28
29
30

P[0][0]=1.000000
```

```

P[0][0]=1.000000
P[0][1]=1.000000
P[0][2]=7.000000
P[0][3]=25.000000
P[0][4]=61.000000
P[0][5]=121.000000
P[1][0]=217.000000
P[1][1]=343.000000
P[1][2]=511.000000
P[1][3]=727.000000
P[1][4]=997.000000
P[1][5]=1327.000000
P[2][0]=1729.000000
P[2][1]=2197.000000
P[2][2]=2743.000000
P[2][3]=3373.000000
P[2][4]=4093.000000
P[2][5]=4909.000000
P[3][0]=5833.000000
P[3][1]=6859.000000
P[3][2]=7999.000000
P[3][3]=9259.000000
P[3][4]=10645.000000
P[3][5]=12163.000000
P[4][0]=13825.000000
P[4][1]=1.000000
P[4][2]=17575.000000
P[4][3]=19681.000000
P[4][4]=21949.000000
P[4][5]=24385.000000
Process returned 0 (0x0)   execution time : 60.850 s
Press any key to continue.

```

83)

2) Вычислить

$$R = f(n \cdot m, k) + f^2(n \cdot k, n) - f^3(m, n),$$

при $k = 3.6$, $m = 2.2$, $n = 5$ если

$$f(a, b) = \sin^2(\log_{10}(a \cdot b)) + \frac{1}{\sqrt{a^2 + b^2}}.$$

3)

$$R = f(n \cdot m, k) + f^2(n \cdot k, n) - f^3(m, n),$$

при $k = 3.6$, $m = 2.2$, $n = 5$ если

$$f(a, b) = \sin^2(\log_{10}(a \cdot b)) + \frac{1}{\sqrt{a^2 + b^2}}.$$

4)

Имя	Смысл	Тип
f	Функция, рассчитывающая f	double f(double x, double y)
x	Переменная функции	double
y	Переменная функции	double
z	Значение функции	double
z1	Переменная функции	double
z2	Переменная функции	double
z3	Переменная функции	double
z4	Переменная функции	double
z5	Переменная функции	double
k	Константы	double
m	Константы	double
n	Константы	double
l	Промежуточная переменная	double
p	Промежуточная переменная	double
g1	Промежуточная переменная	double
g2	Промежуточная переменная	double
g3	Промежуточная переменная	double
S	Значение выражения	double

5)

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>

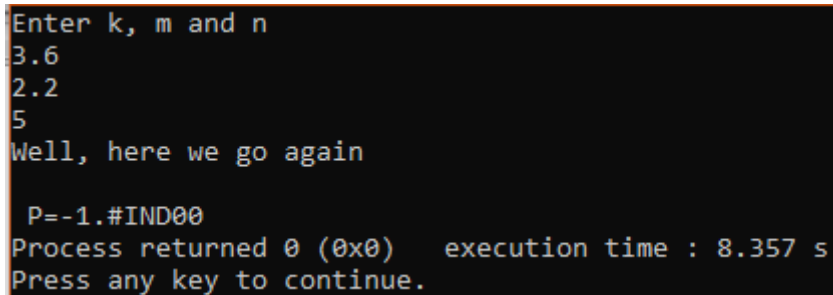
double f(double x, double y)
{
    double z,z1,z2,z3,z4,z5;
    z1=logf(x*y);
    z2=sinf(z1);
    z3=pow(z2,2);
    z4=x*x+y*y;
    z5=sqrtf(z4);
    z=z3+(1/z5);
    return(z);
}
```

```

main()
{
double k,m,n,l,p,g1,g2,g3;
double S;
S=0;
printf("Enter k, m and n \n");
scanf("%f",&k);
scanf("%f",&m);
scanf("%f",&n);
printf("Well, here we go again \n");
l=n*m;
p=n*k;
g1=f(l,k);
g2=f(p,n);
g3=f(m,n);
S=g1+pow(g2,2)-pow(g3,3);
printf("\n P=%f",S);
return 0;
}

```

6)



```

Enter k, m and n
3.6
2.2
5
Well, here we go again

P=-1.#IND00
Process returned 0 (0x0)   execution time : 8.357 s
Press any key to continue.

```

8.4)

2) Определить среднее значение для элементов, находящихся ниже побочной диагонали матрицы $S[5][5]$.

3)

```

for (i=0;i<5;i++)
{
    for(j=0;j<5;j++)
    {
        if (i>=j)
        {
            a+=S[i][j];
            b+=1;
        }
    }
}
L=a/b;

```

4)

Имя	Смысл	Тип
i	Индекс Элемента массива	Int
j	Индекс Элемента массива	Int
a	Промежуточная переменная	Int
b	Промежуточная переменная	Int
L	Среднее значение	Double
S	Матрица	Массив элементов типа int
S[i][j]	Элемент матрицы	Int

5)

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main()
{
    int i,j,a,b;
    double L;
    int S[5][5];
    a=0;
    b=0;

```

```
printf("Enter matrix S\n");
for (i=0;i<5;i++)
{
    for (j=0;j<5;j++)
    {
        scanf("%d\n",&S[i][j]);
    }
}
printf("Well, here we go again \n");
for (i=0;i<5;i++)
{
    for(j=0;j<5;j++)
    {
        if (i>=j)
        {
            a+=S[i][j];
            b+=1;
        }
    }
}
L=a/b;
printf("%f",L);
return 0;
}
```

6)

```
Enter matrix S
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
0
Well, here we go again
15.000000
Process returned 0 (0x0)   execution time : 23.944 s
```

85)

2) Даны две квадратные матрицы A и B размером $N \times N$ каждая. Напечатать ту из них, которая имеет минимальный «след» (т.е. сумму элементов главной диагонали). Математически это обозначается $\text{Tr}(A)$ (англ. - trace) или $\text{Sp}(A)$ (нем. - spur). При решении создать функцию для нахождения следа матрицы и функцию печати матрицы.

3)

```
int Tr(int x, int y, int z)
```

```
{
    if (x==y)
    {
        return z;
    }
    else{
```

```

return 0;

}

}

```

4)

Имя	Смысл	Тип
Tr	Функция, заменяющая элемент массива	int Tr(int x, int y, int z)
x	Переменная функции	int
y	Переменная функции	int
z	Значение, возвращённое функцией	int
**S	Матрица 1	Массив элементов типа int
**F	Матрица 2	Массив элементов типа int
i	Индекс элемента массива	int
j	Индекс элемента массива	int
a	Промежуточная переменная	int
b	Промежуточная переменная	int
N	Размер матрицы	int
TS	След матрицы S	int
TF	След матрицы F	int
S[i][j]	Элемент матрицы 1	int
F[i][j]	Элемент матрицы 2	int

5)

```

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int Tr(int x, int y, int z)
{
    if (x==y)
    {
        return z;
    }
    else{
        return 0;
    }
}

```

```

}
main()
{
int **S=NULL;
int **F=NULL;
int i,j,a,b,N,TS,TF;
a=0;
b=0;
TS=0;
TF=0;
scanf("%d",&N);
S=(int**) malloc(N*sizeof(int*));
for (i=0;i<N;i++)
{
    S[i]=(int*)malloc(N*sizeof(int));
}
F=(int**) malloc(N*sizeof(int*));
for (i=0;i<N;i++)
{
    F[i]=(int*)malloc(N*sizeof(int));
}
printf("Enter matrix S\n");
for (i=0;i<N;i++)
{
    for (j=0;j<N;j++)
    {
        scanf("%d",&S[i][j]);
    }
}
printf("Enter matrix F\n");

```

```

for (i=0;i<N;i++)
{
    for (j=0;j<N;j++)
    {
        scanf("%d",&F[i][j]);
    }
}
printf("Well, here we go again \n");
for (i=0;i<N;i++)
{
    for (j=0;j<N;j++)
    {
        a=Tr(i,j,S[i][j]);
        TS+=a;
    }
}
for (i=0;i<N;i++)
{
    for (j=0;j<N;j++)
    {
        b=Tr(i,j,F[i][j]);
        TF+=b;
    }
}
if (TF<TS)
{
    for (i=0;i<N;i++)
    {
        for (j=0;j<N;j++)
        {

```



```

        printf("%d",F[i][j]);
    }
}
}
else
{
    for (i=0;i<N;i++)
    {
        for (j=0;j<N;j++)
        {
            printf("%d",S[i][j]);
        }
    }
}
return 0;
}

```

6)

```

2
Enter matrix S
1
2
3
4
Enter matrix F
5
6
7
8
Well, here we go again
1234
Process returned 0 (0x0)   execution time : 8.130 s
Press any key to continue.

```