Лабораторная работа №5

1) Тема: Массивы

5.1)

2) Для некоторого числового вектора X, введённого с клавиатуры, вычислить значения вектора Y = X · X (yi = xi · xi — поэлементно).

3) y[i] = x[i] * x[i]

4)

Имя	Смысл	Тип
y	Массив элементов 1	Массив элементов int
x	Массив элементов 2	Массив элементов int
y[i]	Элемент массива 1	Int
x[i]	Элемент массива 2	Int
i	Индекс Элемента массива	Int

5)

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int y[5];
    int x[5];
    for (int i=0; i<5; i+=1)
    {
        scanf ("\n%d", &x[i]);
    }
    for (int i=0; i<5; i+=1)
    {
        y[i]=x[i]*x[i];
        printf("\n%d", y[i]);
    }
    return 0;
}
```

6)

```
1
2
3
4
5

1
4
9
16
25
Process returned 0 (0x0)   execution time : 2.492 s
Press any key to continue.
```

5.2)

2) Для некоторого числового массива X, введённого с клавиатуры, изменить порядок элементов на обратный без привлечения вспомогательного массива и со вспомогательным массивом.

3) a) for (i=0; i<=6/2; i+=1)

{b=x[i];

x[i]=x[6-i-1];

x[6-i-1]=b;}

b) for (i=0; i<=6/2; i+=1)

{y[i]=x[6-i-1];}

4)

Имя	Смысл	Тип
x	Массив 1	Массив элементов типа int
x[i]	Элемент массива 1	Int
i	Индекс массива	Int
b	Промежуточная переменная	Int
l	Промежуточная переменная для индекса	Int
y	Массив 2	Массив элементов типа int
y[i]	Элемент массива 2	Int

5/a)

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int x[6];
    int b,l;
    for(int i=0; i<6;i+=1)
    {
        scanf("%d", &x[i]);
    }
    l=6/2;
    for (int i=0;i<l;i+=1)
    {
        b=x[i];
        x[i]=x[6-i-1];
        x[6-i-1]=b;
    }
    for(int i=0; i<6;i+=1)
    {
        printf("\n%d", x[i]);
    }
     return 0;
}
```

6/a)

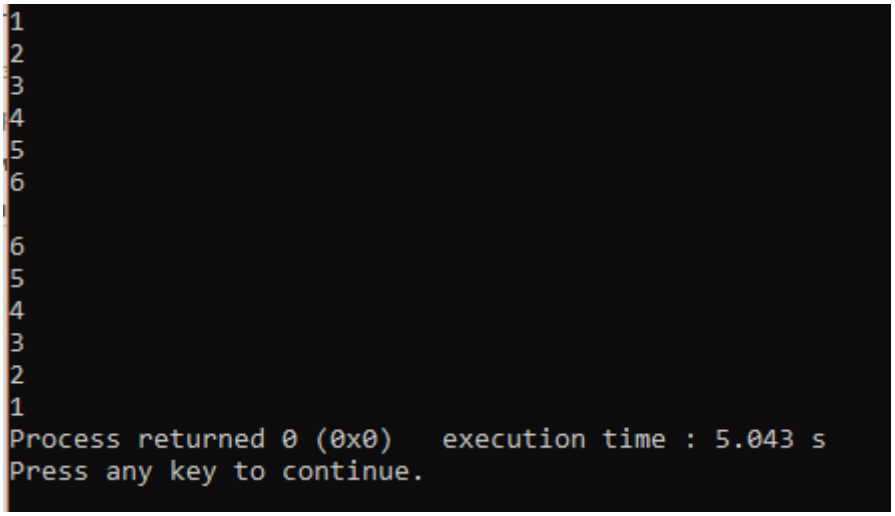
```
1
2
3
4
5
6

6
5
4
3
2
1
Process returned 0 (0x0)   execution time : 4.257 s
Press any key to continue.
```

5/b)

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int x[6];
    int y[6];
    int b,l;
    for(int i=0; i<6;i+=1)
    {
        scanf("%d", &x[i]);
    }
    l=6/2;
    for (int i=0;i<6;i+=1)
    {
        y[i]=x[6-i-1];
    }
    for(int i=0; i<6;i+=1)
    {
        printf("\n%d", y[i]);
    }
     return 0;
}
```

6/b)



5.3)

2) Реализовать различные варианты алгоритма сортировки пузырьком, организовав проходы алгоритма с начала, и с конца массива, а также с двумя противоположными условиями сравнения. В качестве элементов сортировки использовать произвольные массивы чисел. Каждый найденный возможный вариант алгоритма должен приводить к некоторому осмысленному результату сортировки.

3) a) Если (x[i]>x[x+1])

{b=x[i];

x[i]=x[i+1];

x[i+1]=b;}

b) Если (x[i]<x[i-1])

{b=x[i];

x[i]=x[i-1];

x[i-1]=x[1];}

4)

Имя	Смысл	Тип
n	Количество элементов в массиве	Int
x	Массив	Массив элементов типа int
x[i]	Элемент Массива	Int
b	Промежуточная переменная	int
i	Индекс элемента массива	Int
z	Индекс элемента массива	Int

5/a)

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int n;
    scanf("\n%d", &n);
    int x[n];
    int b,l;
    for(int i=0; i<n; i+=1)
    {
        scanf("\n%d",&x[i]);
    }
    for(int i=0; i<n; i+=1)
    {
        printf("\n%d",x[i]);
    }
    printf("\n");
    for (int z=1; z<n;z+=1)
    {


    for (int i=0; i<(n-1); i+=1)
    {
        if (x[i]>x[i+1])
        {
            b=x[i];
            x[i]=x[i+1];
            x[i+1]=b;
        }
    }}
       for(int i=0; i<n; i+=1)
    {
        printf("\n%d",x[i]);
    }
     return 0;
}
```

6/a)

```
5
5
2
3
1
4

5
2
3
1
4

1
2
3
4
5
Process returned 0 (0x0)   execution time : 12.516 s
Press any key to continue.
```

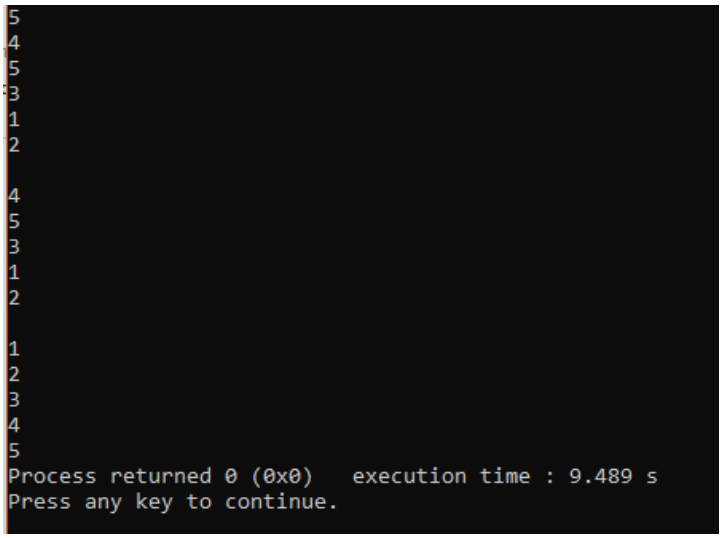
5/b)

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int n;
    scanf("\n%d", &n);
    int x[n];
    int b,l;
    for(int i=0; i<n; i+=1)
    {
        scanf("\n%d",&x[i]);
    }
    for(int i=0; i<n; i+=1)
    {
        printf("\n%d",x[i]);
    }
    printf("\n");
    for (int z=0; z<n;z+=1)
    {


    for (int i=n; i>0; i-=1)
    {
        if (x[i]<x[i-1])
        {
            b=x[i];
            x[i]=x[i-1];
            x[i-1]=b;
        }
    }}
       for(int i=0; i<n; i+=1)
    {
        printf("\n%d",x[i]);
    }
     return 0;
}

```

6/b)



5.4)

2) Реализовать самостоятельно алгоритм сортировки вставками и сравнить его реализацию с полученными реализациями «алгоритма пузырька» в задаче 5.3.

3) for(j=0;j<=10;j++)

{ for (i=0; i<=10;i++)

{if (A[j]<A[i])

{N=A[j];

A[j]=A[i];

A[i]=N;}}

}

4)

Имя	Смысл	Тип
A	Массив	Массив элементов типа int
A[i],A[j]	Элемент Массива	Int
N	Промежуточная переменная	Int
i	Индекс элемента массива	Int
j	Индекс элемента массива	Int

5)

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
    int A[10]={9,8,7,6,5,4,3,2,1,0};
    int N,i,j;
    for (j=0;j<=10;j++)
    {
        for (i=0;i<=10;i++)
        {
            if (A[j]<A[i])
            {
                N=A[j];
                A[j]=A[i];
                A[i]=N;
            }
        }
    }
    for (i=0;i<=10;i++)
    {
        printf("%d\n",A[i]);
    }
    return 0;
}
```

6)

```
0
1
2
3
4
5
6
7
8
9
9

Process returned 0 (0x0)   execution time : 0.013 s
Press any key to continue.
```

5.5)

2) Организовать ввод массива (матрицы) по столбцам.

3) for (i=0; i<2; i++)

{ for (j=0; j<2; j++)

{ scanf(“%d\n”, &A[i][j]) } }

4)

Имя	Смысл	Тип
A	Массив	Массив элементов типа int
A[i][j]	Элемент массива	Int
i	Индекс массива	Int
j	Индекс массива	Int

5)

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
    int A[2][2];
    int N,i,j;
    for (i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            scanf("%d\n", &A[j][i]);
        }
    }
    printf("%d %d \n", A[0][0], A[0][1]);
    printf("%d %d \n", A[1][0], A[1][1]);
    return 0;
}
```

6)

```
1
2
3
4
4
1 3
2 4

Process returned 0 (0x0)   execution time : 4.463 s
Press any key to continue.
```

56)

2) Найти значения матричного многочлена f(A), где

$$f(A) = -2x^2 + 5x + 9,\ A = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

3)

$$f(A) = -2x^2 + 5x + 9,\ A = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

4)

Имя	Смысл	Тип
A	Изначальная матрица	Массив элементов типа int
A[i][j]	Элемент изначальной матрицы	int
B	Матрица, равная E*9	Массив элементов типа int
B[i][j]	Элемент матрицы E*9	Int
Z	Матрица, равная -2*A^2	Массив элементов типа int
Z[i][j]	Элемент матрицы, равной -2*A^2	Int
E	Матрица, равная A*5	Массив элементов типа int
E[i][j]	Элемент матрицы, равной A*5	Int
L	Значение матричного многочлена (матрица)	Массив элементов типа int
L[i][j]	Элемент значения матричного многочлена (матрицы)	Int
i	Индекс матриц	Int
j	Индекс матриц	int

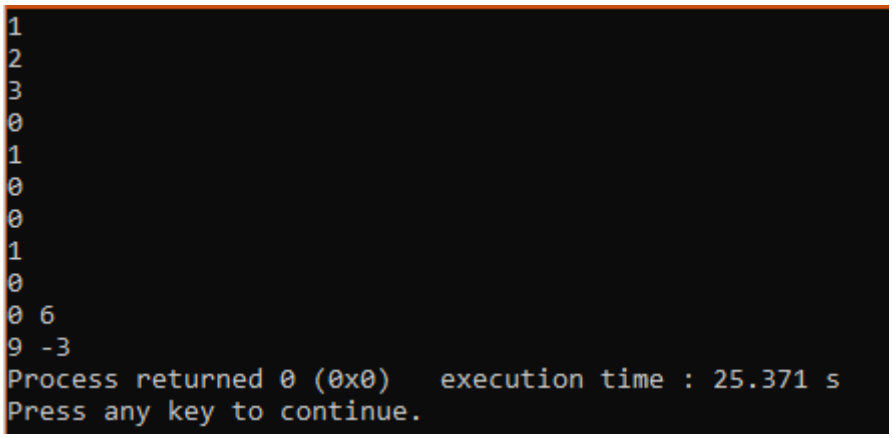
5)

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
   int A[2][2],B[2][2],Z[2][2],E[2][2],L[2][2];
   int N,i,j;
   for (i=0;i<2;i++)
   {
      for(j=0;j<2;j++)
      {
         scanf("%d\n", &A[i][j]);
      }
   }
   for (i=0;i<2;i++)
   {
      for(j=0;j<2;j++)
      {
         scanf("%d\n", &B[i][j]);
```

```
            B[i][j]=B[i][j]*9;
        }
    }
    for (i=0;i<2;i++)
    {
        for (j=0;j<2;j++)
        {
            Z[i][j]=(A[i][0]*A[0][j]+A[i][1]*A[1][j])*(-2);
        }
    }

    for (i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            E[i][j]=A[i][j]*5;
        }
    }
    for (i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            L[i][j]=Z[i][j]+E[i][j]+B[i][j];
        }
    }
printf("%d %d",L[0][0], L[0][1]);
printf("\n%d %d",L[1][0],L[1][1]);
    return 0;
}
```

6)



5.7)

2) Транспонировать матрицу

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

3)

```
for (i=0;i<3;i++)
{
   for(j=0;j<2;j++)
   {
      B[i][j]=A[j][i];
   }
}
```

4)

Имя	Смысл	Тип
A	Начальная матрица	Массив элементов типа int
A[i][j]	Элемент начальной матрицы	Int
B	Транспонированная матрица	Массив элементов типа int
B[i][j]	Элемент транспонированной матрицы	Int
i	Индекс элемента	Int
j	Индекс элемента	Int

5)

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
    int A[2][3],B[3][2];
    int i,j;
    for (i=0;i<2;i++)
    {
        for (j=0;j<3;j++)
        {
            scanf("%d\n",&A[i][j]);
        }
    }
        for (i=0;i<3;i++)
    {
        for(j=0;j<2;j++)
        {
            B[i][j]=A[j][i];
        }
    }
printf(" %d %d",B[0][0],B[0][1]);
printf("\n %d %d",B[1][0],B[1][1]);
printf("\n %d %d",B[2][0],B[2][1]);
return 0;
}
```

6)

```
1
2
3
4
5
6
1
 1 4
 2 5
 3 6
Process returned 0 (0x0)   execution time : 4.509 s
Press any key to continue.
```

5.8) -

5.9)

2) Найти произведения матриц (AB) · C и A · (BC)

$$A = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 & 0 \\ -3 & 1 \end{bmatrix}, C = \begin{bmatrix} 3 & -1 \\ 2 & 3 \end{bmatrix}$$

3)

$$A = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 & 0 \\ -3 & 1 \end{bmatrix}, C = \begin{bmatrix} 3 & -1 \\ 2 & 3 \end{bmatrix}$$

Найти (AB) · C и A · (BC)

4)

Имя	Смысл	Тип
A	Матрица A	Массив элементов типа int
A[i][j]	Элемент матрицы A	Int
B	Матрица B	Массив элементов типа int
B[i][j]	Элемент матрицы B	Int
C	Матрица C	Массив элементов типа int
C[i][j]	Элемент матрицы C	Int
E	Матрица промежуточного произведения	Массив элементов типа int
E[i][j]	Элемент матрицы промежуточного произведения	Int
L	Результат произведения (матрица)	Массив элементов типа int
L[i][j]	Элемент результата произведения (матрицы)	Int
i	Индекс элемента	Int
j	Индекс элемента	Int

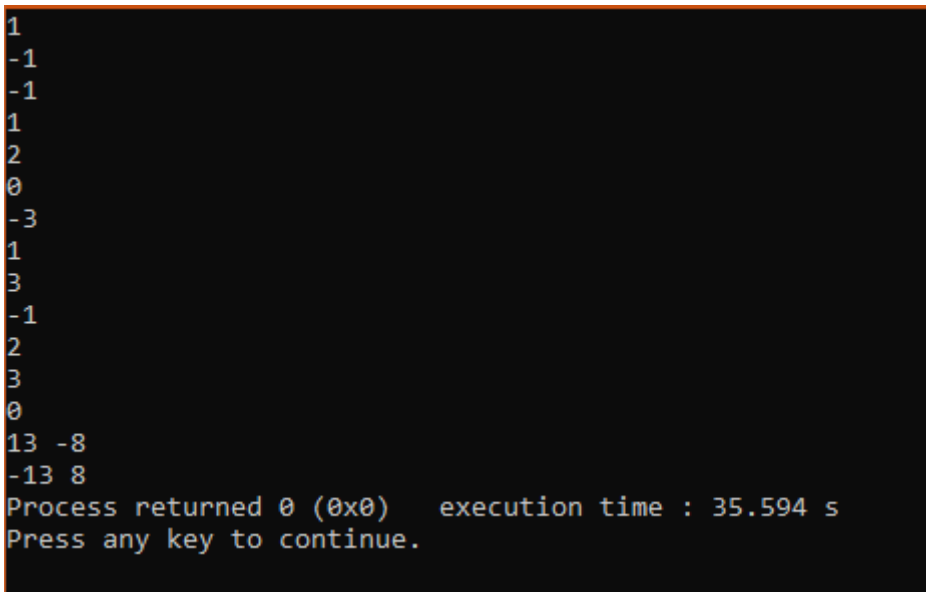
5/a)

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
   int A[2][2],B[2][2],C[2][2],E[2][2],L[2][2];
   int N,i,j;
   for (i=0;i<2;i++)
   {
      for(j=0;j<2;j++)
      {
         scanf("%d\n", &A[i][j]);
      }
   }
    for (i=0;i<2;i++)
   {
      for(j=0;j<2;j++)
      {
```

```
            scanf("%d\n", &B[i][j]);
        }
    }
     for (i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            scanf("%d\n", &C[i][j]);
        }
    }
    for (i=0;i<2;i++)
    {
        for (j=0;j<2;j++)
        {
            E[i][j]=(A[i][0]*B[0][j]+A[i][1]*B[1][j]);
        }
    }


    for (i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            L[i][j]=(E[i][0]*C[0][j]+E[i][1]*C[1][j]);
        }
    }
printf("%d %d",L[0][0], L[0][1]);
printf("\n%d %d",L[1][0],L[1][1]);
    return 0;
}
```

6/a)



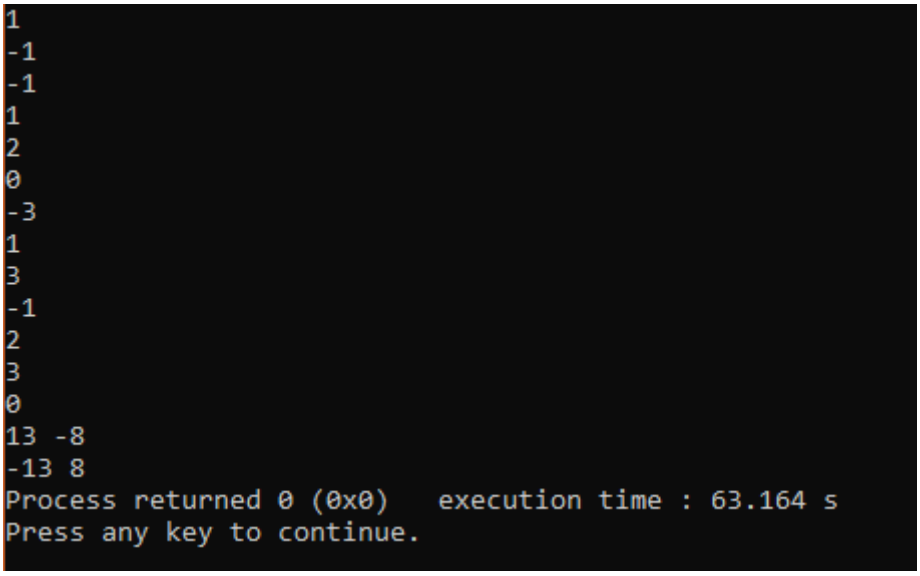
5/b)

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
   int A[2][2],B[2][2],C[2][2],E[2][2],L[2][2];
   int N,i,j;
   for (i=0;i<2;i++)
   {
      for(j=0;j<2;j++)
      {
         scanf("%d\n", &A[i][j]);
      }
   }
    for (i=0;i<2;i++)
   {
      for(j=0;j<2;j++)
      {
         scanf("%d\n", &B[i][j]);
      }
   }
```

```
 for (i=0;i<2;i++)
{
   for(j=0;j<2;j++)
   {
      scanf("%d\n", &C[i][j]);
   }
}
for (i=0;i<2;i++)
{
   for (j=0;j<2;j++)
   {
      E[i][j]=(B[i][0]*C[0][j]+B[i][1]*C[1][j]);
   }
}


for (i=0;i<2;i++)
{
   for(j=0;j<2;j++)
   {
      L[i][j]=(A[i][0]*E[0][j]+A[i][1]*E[1][j]);
   }
}
printf("%d %d",L[0][0], L[0][1]);
printf("\n%d %d",L[1][0],L[1][1]);
   return 0;
}
```

6/b)



5.10)

2) Преобразовать исходную матрицу так, чтобы первый элемент каждой строки был заменён средним арифметическим элементов этой строки.

3)

```
for (i=0;i<3;i++)
{
    for (j=0;j<3;j++)
    {
        B[i]+=A[i][j];
    }
}
A[0][0]=B[0]/3;
A[1][0]=B[1]/3;
A[2][0]=B[2]/3;
```

4)

Имя	Смысл	Тип
A	Матрица	Массив элементов типа int
A[i][j]	Элемент матрицы	Int
B	Массив средних значений	Массив элементов типа int
B[i][j]	Элемент массива средних значений	Int
i	Индекс элемента массива	Int
j	Индекс элемента массива	Int

5)

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
    int A[3][3],B[3]={0,0,0};
    int N,i,j;
    for (i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d\n", &A[i][j]);
        }
    }
     for (i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            B[i]+=A[i][j];
        }
    }
    A[0][0]=B[0]/3;
    A[1][0]=B[1]/3;
    A[2][0]=B[2]/3;
    printf("%d %d %d",A[0][0],A[0][1],A[0][2]);
    printf("\n%d %d %d",A[1][0],A[1][1],A[1][2]);
    printf("\n%d %d %d",A[2][0],A[2][1],A[2][2]);
    return 0;
}
```

6)

```
1
2
3
4
5
6
7
8
9
0
2 2 3
5 5 6
8 8 9
Process returned 0 (0x0)   execution time : 9.796 s
Press any key to continue.
```

6.1)

2) Создать динамический одномерный массив целых чисел F размерности 12 и заполнить его положительными и отрицательными числами.

$$T = \frac{R + Q + S}{R \cdot Q \cdot S + 2},$$

где R — сумма отрицательных элементов F, Q — отрицательный элемент массива, S — произведение положительных элементов массива F.

3)

$$T = \frac{R + Q + S}{R \cdot Q \cdot S + 2},$$

где R — сумма отрицательных элементов F, Q — отрицательный элемент массива, S — произведение положительных элементов массива F.

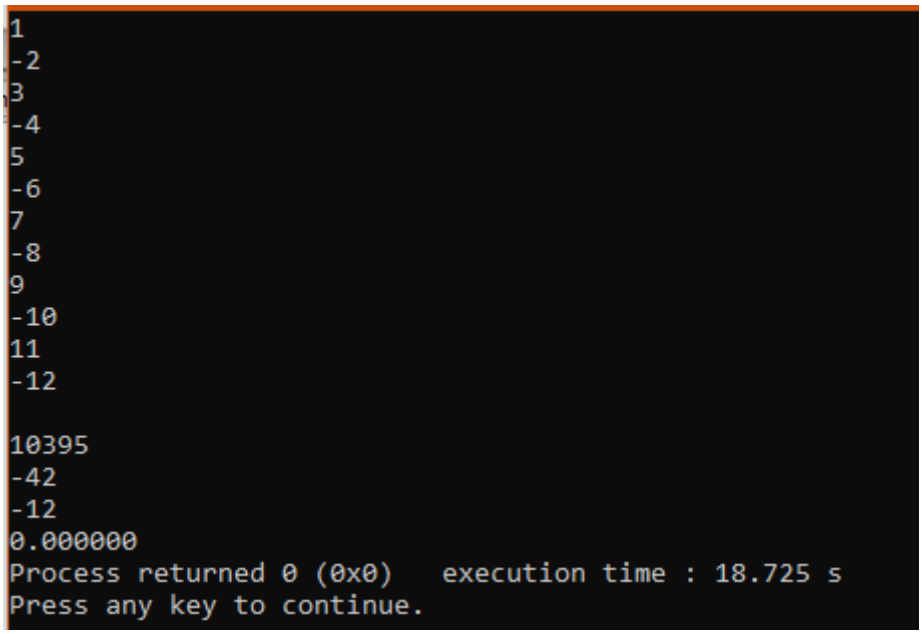
4)

Имя	Смысл	Тип
*piBuffer	Динамический массив F	Int *
piBuffer[i]	Элемент динамического массива F	Int
k	Размер массива	Int
i	Индекс массива	Int
S	Сумма R отрицательных элементов	Int
P	Произведение S положительных элементов	Int
L	Q отрицательный элемент	Int
T	Решение выражения	Double

5)

```
#include <math.h>
int main()
{
    int *piBuffer=NULL;
    int k=12;
    int i=1;
    int S,P,L;
    double T;
    S=0;
    P=1;
    piBuffer=(int *) malloc(k*sizeof(int));
    if (piBuffer==NULL)
    {
        return EXIT_FAILURE;
    }
    for (i=1;i<=k;i++)
        {
        scanf("\n%d",&piBuffer[i]);
        if (piBuffer[i]>0)
        {
            P=P*piBuffer[i];
        }
        if (piBuffer[i]<0)
        {
            S=S+piBuffer[i];
            L=piBuffer[i];
        }
        }
        printf("\n%d\n%d\n%d",P, S, L);
        T=(S+L+P)/(S*L*P+2);
        printf("\n%f",T);
        return 0;
}
```

6)



6.2)

2) Дан одномерный массив целых чисел A размера 12. Вычислить:

$$Y = (U + T) \cdot (S + 2),$$

где S — количество элементов массива A с нечётными индексами, T — наибольший по модулю элемент массива A, U — сумма отрицательных элементов массива A.

3)

$$Y = (U + T) \cdot (S + 2),$$

где S — количество элементов массива A с нечётными индексами, T — наибольший по модулю элемент массива A, U — сумма отрицательных элементов массива A.

4)

Имя	Смысл	Тип
*piBuffer	Динамический массив A	Int *
piBuffer[i]	Элемент динамического массива A	Int
k	Размер массива	Int
i	Индекс массива	Int
S	Количество S элементов с нечётными индексами	Int
U	Сумма U отрицательных элементов	Int
L	Наибольший по модулю элемент T	Int
Y	Решение выражения	Double

5)

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int *piBuffer=NULL;
    int k=12;
    int i=1;
    int U,T,S,L;
    double Y;
    U=0;
    S=0;
    piBuffer=(int *) malloc(k*sizeof(int));
    if (piBuffer==NULL)
    {
        return EXIT_FAILURE;
    }
    for (i=1;i<=k;i++)
        {
        scanf("\n%d",&piBuffer[i]);
        L=piBuffer[i];
        }
        for (i=1;i<=k;i++)
        {
            if (abs(piBuffer[i])>L)
        {
            L=abs(piBuffer[i]);
        }
        if (piBuffer[i]<0)
        {
            U=U+piBuffer[i];
        }
        if (i%2!=0)
        {
            S=S+1;
        }
        }
        printf("\n%d\n%d\n%d",U, S, L);
        Y=(U+L)*(S+2);
        printf("\n%f",Y);
        return 0;
```

6)

```
1
-2
3
-4
5
-6
7
-8
9
-10
11
-12

-42
6
12
-240.000000
Process returned 0 (0x0)   execution time : 19.151 s
Press any key to continue.
```

6.3)

2) Напишите программу для вычисления пересечения двух конечных множеств (наборов) A и B целых чисел одинакового размера с использованием динамических массивов. В качестве множества A можно взять первые 12 чисел ряда Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144. В качестве множества B можно взять первые 12 чисел последовательности Падована: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16. Дублирующиеся значения можно исключать.

3)

```
for (i=0;i<k;i++)
{
   for (j=0;j<k;j++)
   {
      if(A[i]==B[j])
      {
         if (B[j]!=l)
         {
            C[z]=B[j];
            z++;
            l=B[j];
         }
      }
   }
}
```

4)

Имя	Смысл	Тип
*A	Динамический массив чисел Фибоначчи	Int *
A[i]	Элемент массива чисел Фибоначчи	Int
*B	Динамический массив чисел Падована	Int *
B[i]	Элемент массива чисел Падована	Int
*C	Массив пересечения множеств	Int *
C[i]	Элемент массива пересечения множеств	Int
k	Размер массивов чисел Фибоначчи, чисел Падована	Int
i	Индекс элементов массивов	Int
j	Индекс элементов массивов	Int
o	Размер массива пересечения множеств	Int
z	Индекс элемента массива	Int
l	Промежуточная переменная	Int

5)

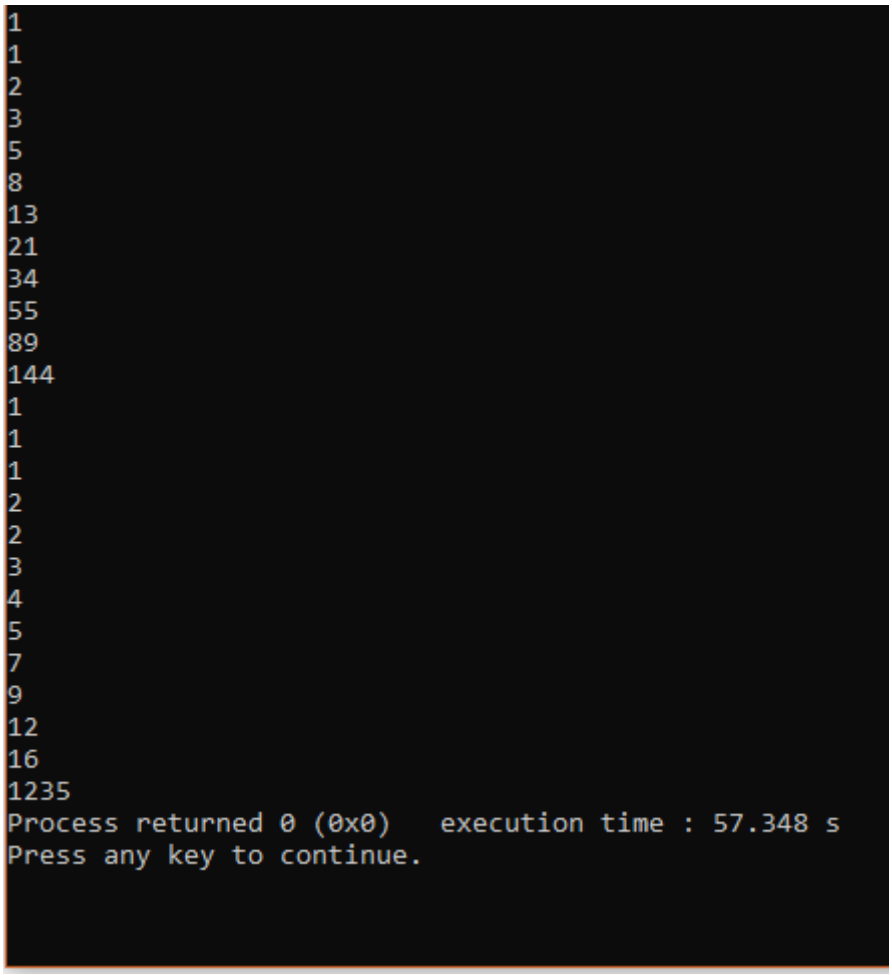
```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <math.h>
main ()
{
int *A=NULL;
int *B=NULL;
int *C=NULL;
int k=12;
int i=1;
int j=1;
int l=0;
int o=0;
int z=0;
A=(int *) malloc(k*sizeof(int));
B=(int*) malloc(k*sizeof(int));
if (A==NULL)
{
   return 1;
}
if (B==NULL)
{
   return 1;
}
for (i=0;i<k;i++)
{
   scanf("%d",&A[i]);
}
for (i=0;i<k;i++)
{
   scanf("%d",&B[i]);
}
for (i=0;i<k;i++)
{
   for (j=0;j<k;j++)
```

```
    {
        if(A[i]==B[j])
        {
            if (B[j]!=l)
            {
                o++;
                l=B[j];
            }
        }
    }
}
C=(int *) malloc(o*sizeof(int));
if (C==NULL)
{
    return 1;
}
l=0;
for (i=0;i<k;i++)
{
    for (j=0;j<k;j++)
    {
        if(A[i]==B[j])
        {
            if (B[j]!=l)
            {
                C[z]=B[j];
                z++;
                l=B[j];
            }
        }
    }
}
for (i=0;i<o;i++)
{
```

```
    printf("%d",C[i]);
}
return 0;
}
```

6)



6.4)

2) Выделить динамически память под некоторую матрицу A размерности M × N и заполнить её произвольными числами. Сократить правильно размер этой матрицы, удалив из неё одну выбранную строку, освободив от неё также и память, используя указатели. После удаления строки в матрице AM×N должна быть возможность обхода всех элементов «новой» матрицы AM−1×N таким же способом, что и изначальной матрицы. Распечатать (используя циклы) матрицу до удаления строки и после удаления.

3)

```
scanf("%d",&x);
z=i-1;
for (k=0;k<i;k++)
{
   for (l=0;l<j;l++)
```

```
    {
        if (k!=x)
        {
            B[o][l]=A[k][l];
            if (l==j-1)
            {
                o+=1;
            }
        }
    }
}
free(A[x]);
```

4)

Имя	Смысл	Тип
A	Начальная матрица	Int **
A[i][j]	Элемент начальной матрицы	Int
B	Конечная матрица	Int **
B[i][j]	Элемент конечной матрицы	Int
k	Индекс элемента	Int
i	Кол-во строк матрицы	Int
j	Кол-во столбцов матрицы	Int
l	Индекс матрицы	Int
o	Индекс конечной матрицы	Int
z	Кол-во строк конечной матрицы	Int
x	Номер удаляемой строки	Int

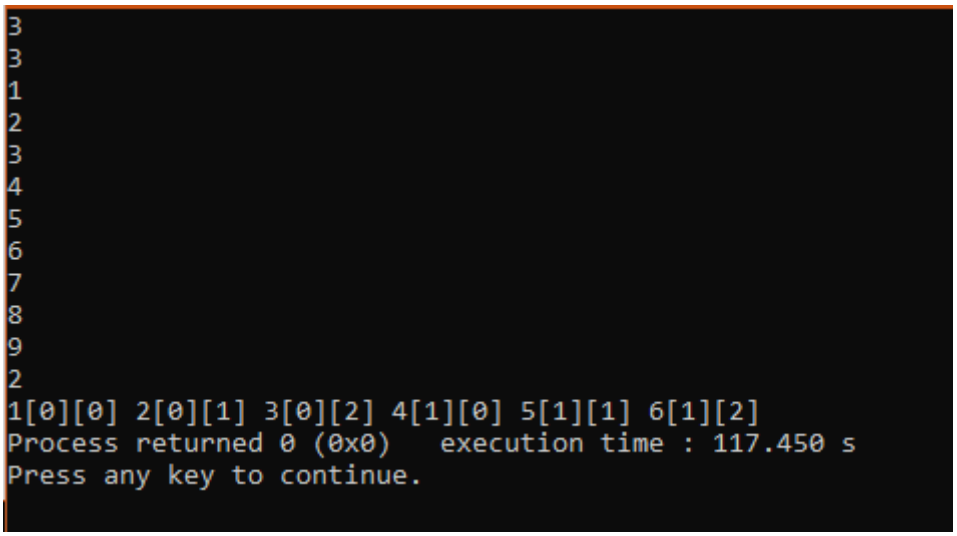
5)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main ()
{
int **A=NULL;
int **B=NULL;
int *C=NULL;
int k=0;
int i;
int j;
```

```
int l=0;
int o=0;
int z=0;
int x;
scanf("%d\n%d", &i, &j);
z=i;
A=(int **) malloc(i*sizeof(int*));
if (A==NULL)
{
   return 1;
}
for (k=0;k<i;k++)
{
   A[k]=(int *) malloc(j*sizeof(int));
}
for (k=0;k<i;k++)
{
   for (l=0;l<j;l++)
   {
      scanf("%d",&A[k][l]);
   }
}
scanf("%d",&x);
z=i-1;
for (k=0;k<i;k++)
{
   for (l=0;l<j;l++)
   {
      if (k==x)
      {
         A[k][l]=0;
      }
   }
}
```

```
B=(int**) malloc(z*sizeof(int*));
if (B==NULL)
{
   return 1;
}
for (k=0;k<z;k++)
{
   B[k]=(int *) malloc(j*sizeof(int));
}
for (k=0;k<i;k++)
{
   for (l=0;l<j;l++)
   {
      if (k!=x)
      {
         B[o][l]=A[k][l];
         if (l==j-1)
         {
            o+=1;
         }
      }
   }
}
for (k=0;k<z;k++)
{
   for (l=0;l<j;l++)
   {
      printf("%d[%d][%d] ",B[k][l], k, l);
   }
}
free(A[x]);
return 0;
}
```

6)



6.5) –

6.6)

2) Напишите программу, в которой создаётся квадратная матрица, заполненная нулями и единицами. Единичные значения у тех элементов, для которых сумма индексов является нечётным числом. Нулевые значения у тех элементов, для которых сумма индексов является чётным числом.

3)

```
 for (k=0;k<i;k++)
{
   for (l=0;l<j;l++)
   {
      z=(k+l)%2;
      if(z>0)
      {
       A[k][l]=1;
      }
      else A[k][l]=0;
   }
}
```

4)

Имя	Смысл	Тип
**A	Матрица	**Int
k	Индекс элемента	Int
l	Индекс элемента	Int
i	Кол-во строк	Int
j	Кол-во столбцов	Int
z	Промежуточная переменная	Int
A[k][l]	Элемент матрицы	Int

5)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main ()
{
int **A=NULL;
int k=0;
int i;
int j;
int l=0;
int z=0;
scanf("%d\n%d", &i, &j);
A=(int **) malloc(i*sizeof(int*));
if (A==NULL)
{
   return 1;
}
for (k=0;k<i;k++)
{
   A[k]=(int *) malloc(j*sizeof(int));
}
for (k=0;k<i;k++)
{
   for (l=0;l<j;l++)
   {
```

```
        z=(k+l)%2;
        if(z>0)
        {
         A[k][l]=1;
        }
        else A[k][l]=0;
    }
}


for (k=0;k<i;k++)
{
    for (l=0;l<j;l++)
    {
        printf("%d[%d][%d] ",A[k][l], k, l);
    }
}
return 0;
}
```

6)