

### III писмени задатак

1. Са стандардног улаза се уносе степен  $n$  и реални коефицијенти полинома редом  $a[n], a[n-1], \dots, a[1], a[0]$  тако да полином се представља у облику израза  $y = a[n] \cdot x^n + a[n-1] \cdot x^{n-1} + \dots + x \cdot a[1] + a[0]$ .

Напишите програм који израчунава вредност тог полинома у  $k$  равномерно распоређених тачака интервала  $[p, q]$ . У првој линији стандардног улаза унети степен полинома  $n$  ( $2 < n < 9$ ), у следећих  $n+1$  линија реалне вредности коефицијената полинома, затим, у наредној линији  $k$  ( $2 < k < 40$ )-број равномерно распоређених тачака на интервалу  $[p, q]$ , у наредној линији реалну вредност  $p$ -почетак интервала, и у наредној линији реална вредност  $q$ -крај интервала. У  $k$  линија исписати вредност полинома у равномерно распоређеним тачакама интервала  $[p, q]$  заокружену на две децимале.

HINT:

```
cout << fixed << showpoint << setprecision(2)
      << VrednostPolinoma(a, n, x) << endl;
```

#### Пример 1

Улаз  
2  
1.0  
2.0  
1.0  
10  
1.0  
10.0

Излаз  
4.00  
9.00  
16.00  
25.00  
36.00  
49.00  
64.00  
81.00  
100.00  
121.00

#### Пример 2

Улаз  
3  
0.54  
1.06  
2.63  
3.42  
7  
-2.00  
3.60

Излаз  
-1.92  
1.17  
3.09  
6.48  
13.98  
28.21  
51.82

### Пример 3

Улаз

8  
0.93  
1.34  
2.25  
3.44  
4.79  
5.70  
6.56  
7.29  
8.15  
11  
-1.00  
5.31

Излаз

4.91  
6.14  
10.64  
31.07  
189.63  
1321.71  
7241.89  
30746.04  
106365.98  
313835.32  
817167.61

Решење:

Вредност полинома  $y=a[n]*x^n+a[n-1]*x^{n-1}+ \dots +x*a[1]+a[0]$  се може израчунати без коришћења операције степеновања ако се користи Хорнерова шема.

Генерисање  $k$  равномерних тачака у интервалу  $[p, q]$  описујемо формулом  $p + i*h$  за  $i=0, 1, \dots, k-1$  и  $h=(q-p)/(k-1)$

```
#include <iostream>
#include <iomanip>
using namespace std;
const int N_MAX = 100;

double VrednostPolinoma(double a[], int n, double x) {
    // Hornerova shema
    double y = 0.0;
    for (int i = n; i >= 0; i--)
        y = y*x + a[i];
    return y;
}

int main() {
    // stepen i koeficijenti polinoma
    int n;
    cin >> n;
    double a[N_MAX];
    for (int i = n; i >= 0; i--)
        cin >> a[i];

    // broj tacaka i granice intervala
    int k;
    double p, q;
    cin >> k >> p >> q;
```

```

// duzina intervala
double h = (q - p) / (k - 1);
// tekuca tacka
double x;
int i;
for (i = 0, x = p; i < k; i++, x += h)
    cout << fixed << showpoint << setprecision(2)
        << VrednostPolinoma(a, n, x) << endl;
}

```

2. Напиши програм који исписује елементе матрице спирално, кренувши од горњег левог угла и кружећи у смеру казаљке на сату. Са стандардног улаза учитава се димензија квадратне матрице  $n$ , а затим и елементи матрице. На стандардни излаз исписати низ елемената матрице који се добијају спиралним обиласком.

Пример 1

Улаз

```

5
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

```

Излаз

```

1 2 3 4 5 10 15 20 25 24 23 22 21 16 11 6 7 8 9 14 19 18 17 12 13

```

Пример 2

Улаз

```

4
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15

```

Излаз

```

0 1 2 3 7 11 15 14 13 12 8 4 5 6 10 9

```

Пример 3

Улаз

```

6
0 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 26 27 28 29
30 31 32 33 34 35

```

Излаз

```

0 1 2 3 4 5 11 17 23 29 35 34 33 32 31 30 24 18 12 6 7 8 9 10 16 22 28 27 26 25
19 13 14 15 21 20

```

Решење

Примера ради, посматрајмо матрицу

```

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

```

Спирални испис можемо организовати тако што исписујемо један по један "прстен" матрице, од спољашњег ка унутрашњим. У нашем примеру то су

```
1 2 3 4 5      7 8 9      13
6      10      12 14
11      15      17 18 19
16      20
21 22 23 24 25
```

Сваки прстен је одређен својим редним бројем  $i$  и вредност броја  $i$  се креће од 0 за спољни прстен, па све док је мања од  $n/2$ . Испис елемената сваког прстена (осим последњег, једночланог) организујемо тако што испишемо његове четири ивице. Започињемо од горње.

Ако је димензија матрице  $n$  непарна, тада се у њеном центру налази последњи, једночлани прстен (заправо само један елемент) и он се исписује само тако што се тај један елемент испише (ово је специјални случај који је потребно проверити након исписа свих прстенова).

```
#include <iostream>

using namespace std;

int main() {
    const int MAX = 100;
    int A[MAX][MAX];
    int n;
    cin >> n;
    for (int v = 0; v < n; v++)
        for (int k = 0; k < n; k++)
            cin >> A[v][k];

    for (int i = 0; i < n / 2; i++) {
        for (int k = i; k < n-i; k++)
            cout << A[i][k] << " ";
        for (int v = i + 1; v < n-i; v++)
            cout << A[v][n-i-1] << " ";
        for (int k = n-i-2; k >= i; k--)
            cout << A[n-i-1][k] << " ";
        for (int v = n-i-2; v > i; v--)
            cout << A[v][i] << " ";
    }

    if (n % 2 != 0)
        cout << A[n/2][n/2] << " ";

    cout << endl;

    return 0;
}
```