

## Rezolvări și explicații – set 3

1. Se citește un șir de "n" numere reale. Să se afișeze: suma acestor numere, valoarea maximă din șir.

Rezolvare:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <limits.h> //pentru constanta INT_MIN adica minus infinit
4  int main()
5  {
6      int n,i=0;
7      float a[20],s=0,max=INT_MIN;
8      // prin float a[20] declarăm un șir de numere reale
9      // șirul a are dimensiunea de maxim 20 elemente
10     // în variabila max vom stoca maximul din șir
11     // initial maximul (necunoscut) presupunem ca e minus infinit
12     printf("Introduceți lungimea șirului n=");
13     scanf("%d",&n);
14     for (i=0;i<n;i++) // parcurgem șirul cu indici de la 0
15     {
16         printf("a[%d]=",i);
17         scanf("%f",&a[i]);
18         s+=a[i]; // reactualizăm suma numerelor
19         if (a[i]>max)
20             max=a[i];
21     }
22     printf("Suma numerelor este: %.2f\n",s);
23     printf("Valoarea maximă din șir este: %.2f\n",max);
24     return 0;
25 }

```

Pentru  $n=10$ , iar șirul  $a[]$  având următoarele elemente în memorie:

i	0	1	2	3	4	5	6	7	8	9
a[i]	2.3	5	1	-1.8	3	4	4.7	-2.2	9.1	4.9

Avem:

- Când  $i=0$ , primul element din șir este  $a[0]=2.3$ , suma devine  $s=0+2.3=2.3$  iar maximul  $\max=a[0]=2.3$  (pentru că  $2.3>\text{INT\_MIN}$ )
- Când  $i=1$ , al doilea element din șir este  $a[1]=5$ , suma devine  $s=2.3+5=7.3$  iar maximul  $\max=a[1]=5$  (pentru că  $5>2.3$ )
- Când  $i=2$ , al treilea element din șir este  $a[2]=1$ , suma devine  $s=7.3+1=8.3$  iar maximul rămâne  $\max=5$  (pentru că  $1<5$ )
- Când  $i=3$ , al patrulea element din șir este  $a[3]=-1.8$ , suma devine  $s=8.3-1.8=6.5$  iar maximul rămâne  $\max=5$  (pentru că  $-1.8<5$ )
- Când  $i=4$ , al cincilea element din șir este  $a[4]=3$ , suma devine  $s=6.5+3=9.5$  iar maximul rămâne  $\max=5$  (pentru că  $3<5$ )

- Când  $i=5$ , al șaselea element din șir este  $a[5]=4$ , suma devine  $s=9.5+4=13.5$  iar maximul rămâne  $\max=5$  (pentru că  $4<5$ )
- Când  $i=6$ , al șaptelea element din șir este  $a[6]=4.7$ , suma devine  $s=13.5+4.7=18.2$  iar maximul rămâne  $\max=5$  (pentru că  $4.7<5$ )
- Când  $i=7$ , al optulea element din șir este  $a[7]=-2.2$ , suma devine  $s=18.2-2.2=16$  iar maximul rămâne  $\max=5$  (pentru că  $-2.2<5$ )
- Când  $i=8$ , al nouălea element din șir este  $a[8]=9.1$ , suma devine  $s=16+9.1=25.1$  iar maximul devine  $\max=a[8]=9.1$  (pentru că  $9.1>5$ )
- Când  $i=9$ , al zecelea element din șir este  $a[9]=4.9$ , suma devine  $s=25.1+4.9=30$  iar maximul  $\max$  rămâne  $\max=9.1$  (pentru că  $4.9<9.1$ )

Deci, când vom tipări la consolă valorile variabilelor  $s$  și  $\max$ , vom vedea ca suma este 30 iar maximul este 9.1.

```

C:\Users\paula\OneDrive\Desktop\test\bin\Debug\test.exe
Introduceti lungimea sirului n=10
a[0]=2.3
a[1]=5
a[2]=1
a[3]=-1.8
a[4]=3
a[5]=4
a[6]=4.7
a[7]=-2.2
a[8]=9.1
a[9]=4.9
Suma numerelor este: 30.00
Valoarea maxima din sir este: 9.10
    
```

2. Se citește un șir de  $n+1$  numere reale  $a_0, a_1, \dots, a_n$  reprezentând coeficienții unui polinom de gradul  $n$ . Se citește o valoare reală  $x$ . Să se calculeze valoarea polinomului în punctul  $x$ .

Rezolvare:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  int main()
5  {
6      //variabila "v" va stoca valoare polinomului
7      float x, a[20], v=0, xp;
8      int n, i;
9      printf("n="); scanf("%d", &n);
10     printf("x="); scanf("%f", &x);
11     for (i=0; i<=n; i++) //sunt n+1 coeficienti, de la 0 la inclusiv "n"
12     {
13         //citim coeficientii si calculam valoarea polinomului
14         printf("Coeficient de ordin %d este ", i);
15         scanf("%f", &a[i]);
16         xp = (float) pow(x, n-i); //calculam x la puterea n adica x^n
17         v = v + xp * a[i]; //echivalent cu v += xp * a[i]
18     }
    
```

```

19     printf("\nPolinomul este:\n");
20     for (i=n; i>=0; i--)
21     {
22         printf("%.2f", a[i]);
23         if (i>0) printf("*X^%d+", i);
24         else printf("*X^%d", i);
25     }
26     printf("\nValoarea polinomului este: %.2f\n", v);
27     return 0;
28 }

```

Pentru  $n=4$ ,  $x=-1$ , iar șirul  $a[]$  având următoarele elemente în memorie:

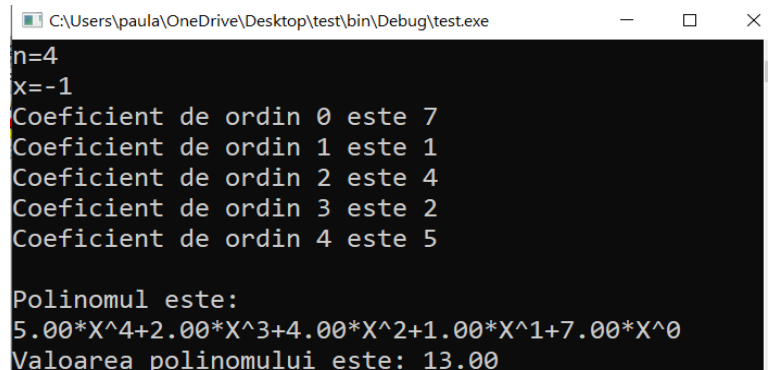
i	0	1	2	3	4
a[i]	7	1	4	2	5

Avem polinomul de forma  $5*X^4+2*X^3+4*X^2+1*X^1+7*X^0$

Valoarea  $v$  a polinomului în punctul  $x=-1$  va fi:

- ❖ Când  $i=0$ , coeficientul de ordin 0 al polinomului este  $a[0]=7$   
Puterea lui  $x$  la 0 este  $x^0=1$   
Valoarea  $v$  se reactualizează  $v=0+1*7=7$
- ❖ Când  $i=1$ , coeficientul de ordin 1 al polinomului este  $a[1]=1$ ,  
Puterea lui  $x$  la 1 este  $x^1=-1$   
Valoarea  $v$  se reactualizează  $v=7+(-1)*1=6$
- ❖ Când  $i=2$ , coeficientul de ordin 2 al polinomului este  $a[2]=4$ ,  
Puterea lui  $x$  la 2 este  $x^2=1$   
Valoarea  $v$  se reactualizează  $v=6+1*4=10$
- ❖ Când  $i=3$ , coeficientul de ordin 3 al polinomului este  $a[3]=2$ ,  
Puterea lui  $x$  la 3 este  $x^3=-1$   
Valoarea  $v$  se reactualizează  $v=10+(-1)*2=8$
- ❖ Când  $i=4$ , coeficientul de ordin 4 al polinomului este  $a[4]=5$ ,  
Puterea lui  $x$  la 4 este  $x^4=1$   
Valoarea  $v$  se reactualizează  $v=8+1*5=13$

În consolă, outputul va fi:



```

C:\Users\paula\OneDrive\Desktop\test\bin\Debug\test.exe
n=4
x=-1
Coeficient de ordin 0 este 7
Coeficient de ordin 1 este 1
Coeficient de ordin 2 este 4
Coeficient de ordin 3 este 2
Coeficient de ordin 4 este 5

Polinomul este:
5.00*X^4+2.00*X^3+4.00*X^2+1.00*X^1+7.00*X^0
Valoarea polinomului este: 13.00

```

3. Se citește o matrice pătratică  $A$  de dimensiune  $n \times n$ . Să se calculeze transpusa matricii  $A$ .

Rezolvare:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int n,a[20][20],i,j,temp;
6      printf("n=");scanf("%d",&n);
7      printf("Introduceti elementele matricii:\n");
8      for(i=0;i<n;i++)
9          for(j=0;j<n;j++) // parcurgem matricea cu indici de la 0
10         {
11             printf("a[%d][%d]=",i,j);
12             scanf("%d",&a[i][j]);
13         }
14     for(i=1;i<n;i++)
15         for(j=0;j<i;j++)
16         {
17             temp=a[i][j];
18             a[i][j]=a[j][i];
19             a[j][i]=temp;
20         }
21     printf("Matricea transpusa este:\n");
22     for(i=0;i<n;i++)
23     {
24         for(j=0;j<n;j++)
25             printf("%d ",a[i][j]);
26         putchar('\n');
27     }
28     return 0;
29 }

```

Pentru  $n=3$  și matricea

$$A = \begin{array}{|c|c|c|} \hline 2 & 2 & 3 \\ \hline 1 & 7 & 9 \\ \hline 1 & 4 & 4 \\ \hline \end{array}$$

Ca să putem calcula transpusa, coloanele matricii initiale devin linii. În rezolvare, ne folosim de 2 indecși  $i$  și  $j$ , unde indexul  $i$  parcurge liniile matricii iar indexul  $j$  parcurge coloanele matricii.

$$A[i,j] = \begin{array}{c} \begin{array}{c} j \rightarrow \\ 0 \quad 1 \quad 2 \end{array} \\ \begin{array}{c} i \downarrow \\ 0 \\ 1 \\ 2 \end{array} \end{array} \begin{array}{|c|c|c|} \hline 2 & 2 & 3 \\ \hline 1 & 7 & 9 \\ \hline 1 & 4 & 4 \\ \hline \end{array}$$

În secvența de cod de la liniile 14-20 construim transpusa astfel:

❖ Când  $i=1, j=0$

- Când  $j=0$  manipulăm elementul  $a[1][0]$  a cărei valoare o copiem într-o variabilă auxiliară/temporară numită  $temp$  (linia 17), pentru că dorim să interschimbăm pe  $a[i][j]$  cu  $a[j][i]$  (liniile 18-19).

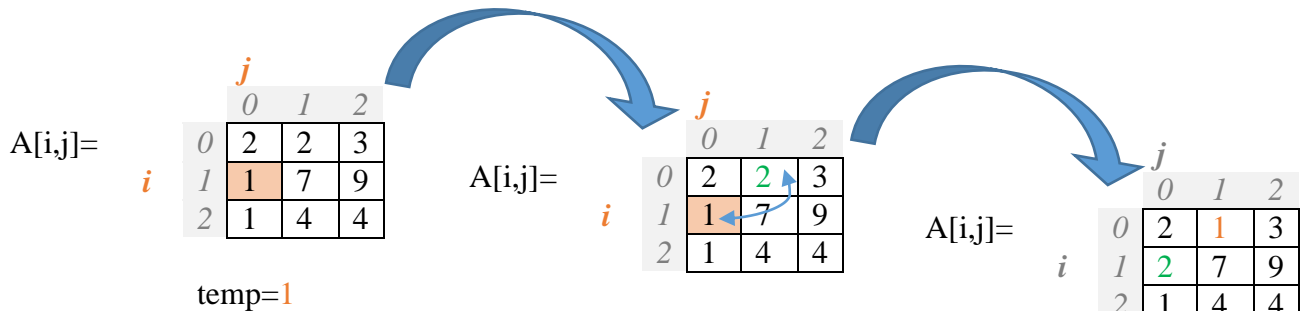
Astfel că,

$temp = a[1][0]$  adică  $temp = 1$

$a[1][0] = a[0][1]$  adică  $a[1][0] = 2$  //interschimbăm pe  $a[1][0]$  cu  $a[0][1]$

$a[0][1] = temp$  adică  $a[0][1] = 1$

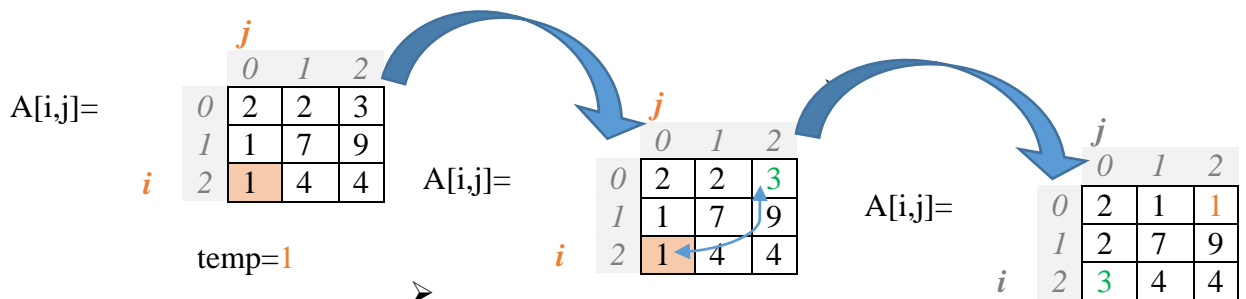
Vizual, are loc aceasta:



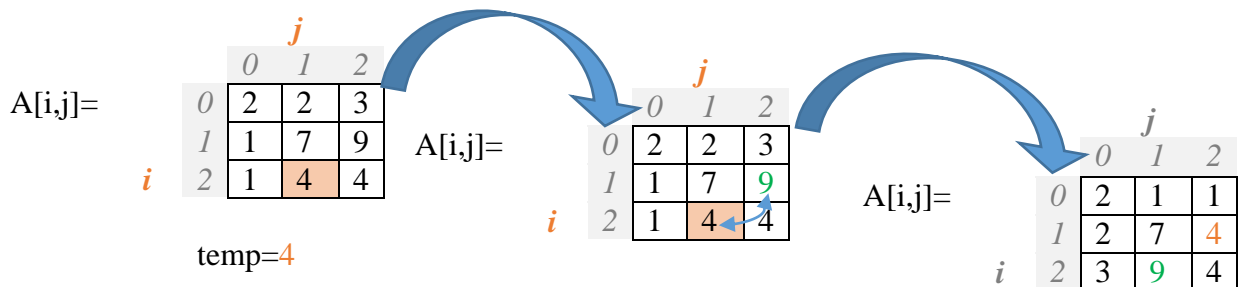
❖ Când  $i=2$

- Când  $j=0$  manipulăm elementul  $a[2][0]$  a cărei valoare o interchimbăm cu  $a[0][2]$  folosind variabila auxiliară  $temp$ .

Avem:



- Când  $j=1$  manipulăm elementul  $a[2][1]$  a cărei valoare o interchimbăm cu  $a[1][2]$  folosind variabila auxiliară  $temp$ .



În consolă outputul va fi:

```
C:\Users\paula\OneDrive\Desktop\test\bin\Debug\test.exe
n=3
Introduceti elementele matricii:
a[0][0]=2
a[0][1]=2
a[0][2]=3
a[1][0]=1
a[1][1]=7
a[1][2]=9
a[2][0]=1
a[2][1]=4
a[2][2]=4
Matricea transpusa este:
2 1 1
2 7 4
3 9 4
```