

Rezolvări și explicații – set 2

1. Se citește un număr natural "n". Să se afișeze secvența: 1 2 2 3 3 3 4 4 4 5 5 5 5

Rezolvare:

Observație: Secvența e cerută pentru cazul particular n=5, însă în rezolvare, algoritmul va funcționa pentru orice valoare a lui "n".

Pentru n=6, spre exemplu, programul trebuie să afișeze secvența:

1 2 2 3 3 3 4 4 4 5 5 5 5 6 6 6 6 6

În secvență observăm că cifra 1 apare 1 dată, cifra 2 de 2 ori, cifra 3 de 3 ori, ș.a.m.d.

Vom folosi un contor "j" care numără de câte ori trebuie să se afișeze un număr "i".

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int i,j,n;
7      printf("n=");scanf("%d",&n);
8      for(i=1;i<=n;++i)
9          for(j=1;j<=i;j++)
10             printf("%d ",i);//tiparim numarul
11     printf("\n");
12     return 0;
13 }
```

2. Se citește un număr natural "n". Să se afișeze secvența: 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5

Rezolvare:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int i,j,n;
7      printf("n=");scanf("%d",&n);
8      for(i=1;i<=n;++i)
9          for(j=1;j<=i;j++)
10             printf("%d ",j);//tiparim contorul
11     printf("\n");
12     return 0;
13 }
```

3. Un număr natural este palindrom dacă citit de la dreapta la stânga rezultă același număr. Să se scrie o funcție care verifică dacă un număr este sau nu palindrom.

Rezolvare:

Spre exemplu, numărul $n=237$ nu este palindrom fiindcă inversul numărului (numărul citit de la dreapta la stânga) este 732.

În schimb, numărul $n=121$ este număr palindrom fiindcă inversul este același număr 121.

Observație: pentru a construi inversul numărului vom folosi algoritmul de ultimă cifră.

Metoda 1 – fără funcție auxiliară

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int n, copie, ninvers=0, uc;
7      printf("n="); scanf("%d", &n);
8      copie=n; // in variabila copie stocam valoarea originala a lui n
9      //in variabila ninvers construim inversul lui n
10     while (n>0)
11     {
12         uc=n%10; // ultima cifra a lui n, la un moment dat de timp
13         ninvers=ninvers*10+uc;
14         n=n/10;
15     }
16     if (ninvers==copie) printf("Numarul %d este palindrom\n", copie);
17     else printf("Numarul %d NU este palindrom\n", copie);
18     return 0;
19 }
```

Metoda 2 – cu funcție auxiliară

Vom crea o funcție generică care pentru un *numar* transmim ca parametru ne va furniza ca rezultat valoarea **1** (true) sau **0** (false): **1** daca numarul este palidrom, **0** daca numărul nu e palindrom.

În rezolvarea de mai jos, funcția *isPalindrom(int numar)* conține instrucțiunile prin care verificăm dacă un numar e palindrom sau nu. În *main()*, în urma citirii lui "n" vom apela această funcție, pentru a verifica dacă numărul introdus de la tastatura este palindrom sau nu.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int isPalindrom (int numar)
4  {
5      int invers=0, uc, copie;
6      copie=numar;
7      while (numar>0)
8      {
9          uc=numar%10;
10         invers=invers*10+uc;
11         numar=numar/10;
12     }
13     if (copie==invers)
14         return 1; //true->numar este palindrom
15     else
16         return 0; //false->numar nu este palindrom
17 }
```

```
17 }
18 int main()
19 {
20     int n, result;
21     printf("Introduceti n="); scanf("%d", &n);
22     result=isPalindrom(n);
23     if (result)
24         printf("Numarul %d este palindrom\n", n);
25     else
26         printf("Numarul %d NU este palindrom\n", n);
27     return 0;
28 }
```