

FUNDAMENTELE PROGRAMĂRII

1. Implementați următoarele funcții:

- funcție care calculează răsturnatul/oglinditul unui număr natural n (răsturnatul numărului 21753 este 35712, iar răsturnatul numărului 1000 este 1);
- funcție care elimină toate cifrele egale cu o cifră c dintr-un număr natural n (dacă $n = 2122753$ și $c=2 \Rightarrow n = 1753$);

Folosind funcția definită la punctul b), scrieți un program care afișează numerele naturale cuprinse între două numere naturale a și b cu proprietatea că:

- sunt palindromuri (de exemplu, 72327);
- nu conțin o cifră c dată;
- sunt formate din cifre egale;
- sunt formate din exact două cifre distincte (de exemplu, 811181).

Rezolvare:

a)

Numărul n	Răsturnatul r al lui n
21753	0
2175	$0 * 10 + 3 = 3$
217	$3 * 10 + 5 = 35$
21	$35 * 10 + 7 = 357$
2	$357 * 10 + 1 = 3571$
0	$3571 * 10 + 2 = 35712$

b)

Numărul n	Variabila aux
2122753	0
212275	$0 * 10 + 3 = 3$
21227	$3 * 10 + 5 = 35$

Numărul n	Variabila aux
2122	$35 * 10 + 7 = 357$
212	357
21	357
2	$357 * 10 + 1 = 3571$
0	3571

Acum trebuie să calculez răsturnatul lui aux = 3571 => răsturnat(aux) = 1753.

Numărul n	Variabila aux
2120027500022	0
2120027500022	Eliminăm cifrele egale cu c de la sfârșitul lui n
21200275000	Eliminăm cifrele egale cu 0 de la sfârșitul lui n și le numărăm
21200275	Eliminăm cifrele egale cu c din n
57001	aux
10075	Răsturnăm aux
10075000	Adăugăm la sfârșit cele 3 zerouri eliminate

Numărul n	Variabila aux (tablou unidimensional)														
212002785002022	2	2	0	2	0	0	5	8	7	2	0	0	2	1	2
	Variabila nr (tablou unidimensional)														
	1	0	0	7	8	5	0	0	0						
	Variabila n									Variabila p					
	0									1000000000					
	$0 + 1 * 1000000000$									1000000000					
	$1000000000 + 0 * 1000000000$									1000000000					
	$1000000000 + 0 * 1000000000$									1000000000					
	$1000000000 + 7 * 100000000 = 1007000000$									1000000000					
	$1007000000 + 8 * 100000000 = 1007800000$														

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

```
unsigned long long int rasturnat(unsigned long long int n)
{
    unsigned long long int r;

    r = 0;
    while(n != 0)
    {
```

```

        r = r * 10 + n % 10;
        n = n / 10;
    }

    return r;
}

void eliminare_cifra(unsigned long long int *n, unsigned int c)
{
    unsigned int aux[20], nr[20], nc, k;
    unsigned long long int p;
    int i;

    nc = 0;
    while(*n != 0)
    {
        aux[nc] = *n%10;
        nc++;
        *n = *n/10;
    }

    k = 0;
    p = 1;
    for(i = nc-1; i >= 0; i--)
        if(aux[i] != c)
        {
            nr[k] = aux[i];
            k++;
            p = p * 10;
        }

    p = p / 10;
    *n = 0;
    for(i = 0; i < k; i++)
    {
        *n = *n + p * nr[i];
        p = p / 10;
    }
}

int main()
{
    unsigned long long int a, b, i, aux;
    unsigned int c, p, q;

    printf("a = ");
    scanf("%I64u", &a);
    printf("b = ");
    scanf("%I64u", &b);

```

```

printf("\nPalindromurile cuprinse intre %I64u si %I64u:\n", a, b);
for(i = a; i <= b; i++)
    if(i == rasturnat(i))
        printf("%I64u ", i);
printf("\n\n");

printf("Cifra c = ");
scanf("%u", &c);
printf("\nNumerele cuprinse intre %I64u si %I64u care nu contin cifra
%u:\n", a, b, c);
for(i = a; i <= b; i++)
{
    aux = i;
    eliminare_cifra(&aux, c);
    if(aux == i)
        printf("%I64u ", i);
}

printf("\n\n");

printf("\nNumerele cuprinse intre %I64u si %I64u care sunt formate din
cifre egale:\n", a, b);
for(i = a; i <= b; i++)
{
    aux = i;
    eliminare_cifra(&aux, 0);
    if(aux == i)
    {
        eliminare_cifra(&aux, aux%10);
        if(aux == 0)
            printf("%I64u ", i);
    }
}

printf("\n\n");

return 0;
}

```