

Subiectul I

1. a)

2.

$$\begin{aligned} & f(102030) \rightarrow f(110203) \rightarrow f(11020) \rightarrow f(102) \rightarrow f(10) \\ & 20 + f(10203) \quad 20 + f(11020) \quad 20 + f(1102) \quad 20 + f(10) \quad 2020 \end{aligned}$$

$$20 + 20 + 20 + 20 + 2020 = 2100$$

c)

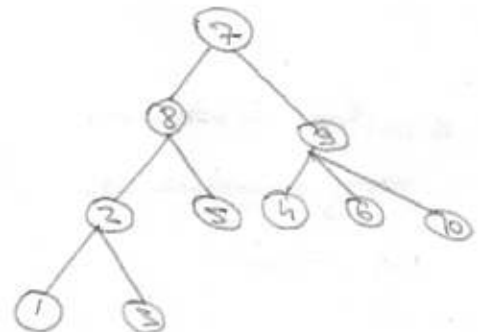
3.

1, 101, 103, 11, 111, 113, 121, 123, 13, 131, 133, 201, 203, 21, ...

12

d) 4. Nodurile cu exact 2 fii: 2, 8, 7

b)



5. $m=20$

$m=100$

Graful este complet \Rightarrow nr. de muchii = $\frac{n(n-1)}{2} = \frac{20 \cdot 19}{2} = 190$

Numărul de muchii ce trebuie adăugate: $190 - m = 90$ muchii

c)

Subiectul II

1. Programul descrie în pseudocod citeste un număr natural n , iar apoi t numere naturale, unde $t = nr.$ de cifre ale lui n , și construiește un număr a cărui ultimă cifră este ultima cifră a primului număr, ~~a~~ ~~doi~~ penultima cifră este penultima cifră din cel de-al doilea număr ș.a.m.d.

În cazul în care un număr nu are suficiente cifre, cifra se va lua cu ea de pe poziția corespunzătoare din numărul n .

a) 2020

b) 9 40

19 341

c.

include <iostream>

using namespace std;

int main()

{
int n, p=1, m=0, k=0;

cin >> n;

while (n != 0)

{

cin >> x;

for (i=1; i<=k; i++) x/=10;

if (x != 0) c = x%10;

else c = m%10;

m = c * p + m;

m /= 10;

p *= 10; k++;

cout << m;

return 0;

}

b)

estete m

$p \leftarrow 1; m \leftarrow 0; l_u \leftarrow 0$

est hmp $m \neq 0$ executo

estete x

$x \leftarrow x/p$

do $x \neq 0$ otuna $c \leftarrow x \% 10$

offfel $c \leftarrow m \% 10$

$m \leftarrow c * p + m$

$m \leftarrow \lfloor m / 10 \rfloor$

$p \leftarrow p * 10; l_u \leftarrow l_u + 1$

este m

2.

struct triunghi

{

struct

{

float x;

float y;

} A, B, C;

} t;

3.

SVICTORIE

Subject II

```

1. void future(int m, int &d, int &p)
{
    d=1, p=0;
    int dv=2, pt;
    while(m!=1)
    {
        if(m%dv==0)
        {
            pt=0;
            while(m%dv==0)
            {
                pt++;
                m/=dv;
            }
            if(pt > p || (pt == p && dv < d))
            {
                d=dv;
                p=pt;
            }
        }
        dv++;
    }
}

```

```

2. #include <iostream>
using namespace std;
int m, h, m[21][401];
int main()
{
    cin >> m >> h;
    for(int i=1; i<=m; i++)
        for(int j=1; j<=m*h; j++)
            m[i][j] = (j-1)/h + i;
    for(int i=1; i<=m; i++)
        for(int j=1; j<=m*h; j++)
            cout << m[i][j] << " ";
    return 0;
}

```

```
#include <fstream>
#include <iostream>
```

```
using namespace std;
```

```
ofstream fout("src.txt");
```

```
int moim()
```

```
{
```

```
    int x, y, c;
```

```
    cin >> x >> y;
```

```
    cout << y << " " << x;
```

```
    while (! (x == 1 && y == 1))
```

```
    {
```

```
        c = 3 * x - y;
```

```
        cout << c << " ";
```

```
        y = x; x = c;
```

```
    }
```

```
    return 0;
```

```
}
```

- 8) Algoritmul se bazează pe determinarea termenului imediat anterior folosindu-se de formula de recurență a sirului ($f_{n-2} = 3f_{n-1} - f_{n-2}$), având o complexitate de timp liniară și complexitate de memorie constantă.