

FUNDAMENTELE PROGRAMĂRII

1. Se citește un număr natural n . Să se afișeze frecvența fiecărei cifre din scrierea sa.

Rezolvare:

```
#include <stdio.h>

int a = 0, b = 0, c = 0, d = 0, e = 0, f = 0, g = 0, h = 0, i = 0, j = 0;

void storeDigit(int digit)
{
    if(digit == 0) a++;
    if(digit == 1) b++;
    if(digit == 2) c++;
    if(digit == 3) d++;
    if(digit == 4) e++;
    if(digit == 5) f++;
    if(digit == 6) g++;
    if(digit == 7) h++;
    if(digit == 8) i++;
    if(digit == 9) j++;
}

void printDigits()
{
    if (a != 0)
    {
        printf("\nCifra 0 are frecventa: %d \n", a);
    }
}
```

```

if (b != 0)
{
    printf("\nCifra 1 are frecventa: %d \n", b);
}

if (c != 0)
{
    printf("\nCifra 2 are frecventa: %d \n", c);
}

if (d != 0)
{
    printf("\nCifra 3 are frecventa: %d \n", d);
}

if (e != 0)
{
    printf("\nCifra 4 are frecventa: %d \n", e);
}

if (f != 0)
{
    printf("\nCifra 5 are frecventa: %d \n", f);
}

if (g != 0)
{
    printf("\nCifra 6 are frecventa: %d \n", g);
}

if (h != 0)
{
    printf("\nCifra 7 are frecventa: %d \n", h);
}

if (i != 0)
{
    printf("\nCifra 8 are frecventa: %d \n", i);
}

if (j != 0)
{
    printf("\nCifra 9 are frecventa: %d \n", j);
}

int main()
{
    unsigned long long int n;
    printf("Introduceti variabila n: ");
    scanf("%llu", &n);

    while(n != 0)
    {
        int digit = n % 10;
        storeDigit(digit);
        n = n / 10;
    }

    printDigits();

    return 0;
}

```

2. Gigel își dorește foarte mult să-și cumpere o jucărie care costă s RON. Pentru a reuși cât mai repede acest lucru, el se hotărăște să depună în pușculița sa, în fiecare zi, câte o sumă de bani (număr natural nenul). Cunoscând sumele depuse de Gigel zilnic, afișați după câte zile Gigel reușește să strângă în pușculiță suma necesară, suma medie zilnică pe care acesta a depus-o în pușculiță, precum și suma care îi rămâne după ce își cumpără jucăria.

Rezolvare:

```
#include<stdio.h>

int main()
{
    int s;
    int zi = 0, total = 0;
    int suma;

    printf("Costul jucariei: ");
    scanf("%d", &s);

    do
    {
        zi++;
        printf("Suma din ziua %d: ", zi);
        scanf("%d", &suma);
        total += suma;
    }
    while(total < s);

    printf("Numar de zile: %d\n", zi);
    printf("Suma medie zilnică depusa: %.2f\n", (double)total/zi);
    printf("Suma ramasa: %d\n", total-s);

    return 0;
}
```

3. Se citește un număr natural nenul n . Să se calculeze suma $S_n = 1 * 2 + 2 * 3 + \dots + (n - 1) * n$.

Rezolvare:

$$S_n = 1 * 2 + 2 * 3 + \dots + (n - 1) * n = \sum_{i=2}^n (i - 1) * i$$

```
#include <stdio.h>

int main()
{
    int n, i, sum = 0;

    printf("Introduceti n: ");
    scanf("%d", &n);

    for (i = 2 ; i <= n; i++)
        sum = sum + (i-1)*i;

    printf("Suma: %d", sum);

    return 0;
}
```

Varianta optimă:

$$\begin{aligned} S_n &= \sum_{i=2}^n i^2 - \sum_{i=2}^n i = \frac{n(n+1)(2n+1)}{6} - 1 - \frac{n(n+1)}{2} + 1 = \frac{n(n+1)(2n-2)}{6} \\ &= \frac{n(n-1)(n+1)}{3} \end{aligned}$$

4. Se citesc două numere întregi a și b . Scrieți un program care calculează a^b .

Rezolvare:

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

int main()
{
    int a, b, i, result;

    printf("Introduceti doua numere, baza si exponent:\n");
    scanf("%d %d", &a, &b);
```

```

if (a == 0 && b == 0)
{
    printf("\n%d la puterea %d este nedefinit\n", a, b);

    //termin fortat programul
    return 0;
}

result = 1;
for (i = 1; i <= abs(b); i++)
    result *= a;

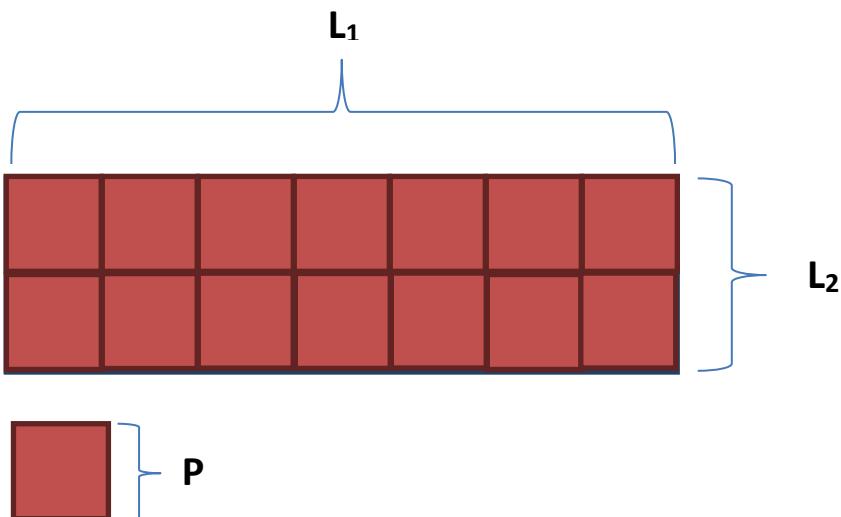
if (b < 0)
    printf("\n%d la puterea %d este egal cu %f\n", a, b, 1.0/result);
else
    printf("\n%d la puterea %d este egal cu %d\n", a, b, result);

return 0;
}

```

5. Un meșter trebuie să paveze întreaga pardoseală a unei bucătării de formă dreptunghiulară, de dimensiune $L_1 \times L_2$ centimetri, cu plăci de gresie pătrate, toate cu aceeași dimensiune. Știind că meșterul nu vrea să taie nici o placă de gresie și vrea să folosească un număr minim de plăci, să se determine dimensiunea plăcilor de gresie de care are nevoie, precum și numărul lor.

Rezolvare:



- meșterul nu vrea să taie nici o placă de gresie

$\begin{array}{l} P \mid L_1 \\ P \mid L_2 \end{array} \Rightarrow P$ este un divizor comun pentru L_1 și L_2

Dacă $L_1 = 140 \text{ cm}$ și $L_2 = 100 \text{ cm} \Rightarrow P \in \{1, 2, 4, 5, 10, 20, 25, 50, 100\} \cap \{1, 2, 4, 5, 7, 10, 14, 20, 28, 35, 70, 140\} \Rightarrow P \in \{1, 2, 4, 5, 10, 20\} \text{ cm}$

- vrea să folosească un număr minim de plăci

Valoarea lui P trebuie să fie maximă $\Rightarrow P = 20 \text{ cm}!$

Numărul de plăci necesare $= \frac{L_1}{P} \times \frac{L_2}{P} = \frac{L_1 \times L_2}{P^2}$

În concluzie, P trebuie să fie c.m.m.d.c.(L_1, L_2)!

Algoritmul lui Euclid pentru calculul c.m.m.d.c.-ului a două numere întregi nenule

c.m.m.d.c.(0, 0) = indefinit

c.m.m.d.c.(x, 0) = x

a = 180, b = 34

a	b	cât	rest
180	34	5	10
34	10	3	4
10	4	2	2
4	2	2	0

cmmdc(a, b) = ultimul rest nenul = ultimul împărțitor

cmmdc(180, 34) = 2

$a = 1.000.000$ și $b = 1 \Rightarrow \text{cmmdc}(a, b) = 1$

a	b	cât	rest
1000000	1	1000000	0

$a = 34, b = 180$

a	b	cât	rest
34	180	0	34
180	34	5	10
...

$180 : 34 = 5$, rest 10

$a = 180, b = 34$

$a = 180 - 34 = 146, b = 34$

$a = 146 - 34 = 112, b = 34$

$a = 112 - 34 = 78, b = 34$

$a = 78 - 34 = 44, b = 34$

$a = 44 - 34 = 10, b = 34$

de 5 ori

6. Scrieți un program care afișează puterile lui 2 aflate într-un interval $[a, b]$. De exemplu, în intervalul $[10,100]$ se găsesc următoarele puteri ale lui 2: 16, 32 și 64.

Rezolvare:

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

```
int main()
{
    int p, a, b;

    printf("a = ");
    scanf("%d", &a);

    printf("b = ");
    scanf("%d", &b);
```

```

p = 1;
while(p <= b)
{
    if(p >= a)
        printf("%d ", p);
    p = p * 2;
}
return 0;
}

```

$1 \leq p \leq b \Leftrightarrow 1 \leq 2^k \leq b$, unde k reprezintă numărul de iterații \Leftrightarrow
 $\log_2 1 \leq \log_2 2^k \leq \log_2 b \Leftrightarrow 0 \leq k \leq \log_2 b \Rightarrow$ numărul de iterații $k \approx \lceil \log_2 b \rceil$

7. Într-o anumită zi a săptămânii, toți cei n studenți ai Facultății de Informatică sunt prezenți la cursuri, însă fiecare într-un anumit interval orar de forma $[a, b]$, unde $a, b \in \mathbb{N}$ și $a < b$. Decanul Facultății dorește să convoace o ședință la care să participe toți studenții. Pentru a-l ajuta, scrieți un program care să determine intervalul orar din ziua respectivă în care sunt prezenți în Facultate toți studenții.

Rezolvare:

```
#include <stdio.h>

int main()
{
    unsigned int n;
    int student, a1, b1, a2, b2, maxA, maxB, ans;

    printf("Vreti sa introduceti manual studenti sau vreti sa folositi functia
random? [0 Manual, !=0 Rand]: ");
    scanf("%d", &ans);
    printf("Introduceti numarul studentilor inscrisi la Facultate: ");
    scanf("%u", &n);

    srand(time(0)); if

        (ans == 0)
    {
        for (student = 0; student<n; student++) {
            if (student == 0) {
Re_insert1:
            printf("Introduceti primul termen (a1) al intervalului [a1 - b1]: ");
            scanf("%u", &a1);
            printf("Introduceti al doilea termen (b1) al intervalului [a1 - b1]: ");
            scanf("%u", &b1);

            if (a1>b1 || a1<8 || b1>20)
            {
                printf("ORAR GRESIT!\n");
                goto Re_insert1;
            }
Re_insert2_1:
            printf("Introduceti primul termen (a2) al intervalului [a2 - b2]: ");
            scanf("%u", &a2);
            printf("Introduceti al doilea termen (b2) al intervalului [a2 - b2]: ");
            scanf("%u", &b2);

            if (a2>b2 || a2<8 || b2>20)
            {
                printf("ORAR GRESIT!\n");
                goto Re_insert2_1;
            }

            if (a1 >= a2)
            {
                maxA = a1;
            }
            else
            {
                maxA = a2;
            }

            if (b1 <= b2)
            {
                maxB = b1;
            }
            else
            {
                maxB = b2;
            }

            printf("\nStudent %d: Interval orar -> [%d - %d]", student+1, a1, b1);
            printf("\nStudent %d: Interval orar -> [%d - %d]", student+2, a2, b2);
        }
    }
}
```

```

if (student >= 2)
{
    Re_insert2_2:
        printf("Introduceti primul termen (a2) al intervalului [a2 - b2]: ");
        scanf("%u", &a2);
        printf("Introduceti al doilea termen (b2) al intervalului [a2 - b2]: ");
        scanf("%u", &b2);

        if (a2>b2)
        {
            printf("ORAR GRESIT!\n");
            goto Re_insert2_2;
        }

        printf("\nStudent %d: Interval orar -> [%d - %d]", student+1, a2, b2);

        if (maxA <= a2)
        {
            maxA = a2;
        }

        if (maxB >= b2)
        {
            maxB = b2;
        }
    if (student == 0)
    {
        student++;
    }

    a1 = a2;
    b1 = b2;

}
else
{
    for (student = 0; student<n; student++)
    if (student == 0)
    {
Re_insert3:
        a1 = (rand() % (8 - 16 + 1)) + 8;
        b1 = (rand() % (16 - 20 + 1)) + 14;

        if (a1>b1 || a1==b1 || b1-a1<2 || b1-a1>8 || (b1-a1)%2!=0)
        {
            goto Re_insert3;
        }
    }

Re_insert4_1:
    a2 = (rand() % (8 - 16 + 1)) + 8;
    b2 = (rand() % (16 - 20 + 1)) + 16;

    if (a2>b2 || a2==b2 || b1-a1<2 || b2-a2>8 || (b2-a2)%2!=0)
    {
        goto Re_insert4_1;
    }

    if (a1 >= a2)
    {
        maxA = a1;
    }
    else
    {
        maxA = a2;
    }

    if (b1 <= b2)
    {
        maxB = b1;
    }
}

```

```

        else
        {
            maxB = b2;
        }

        printf("\nStudent %d: Interval orar -> [%d - %d]", student+1, a1, b1);
        printf("\nStudent %d: Interval orar -> [%d - %d]", student+2, a2, b2);
    }

    if (student >= 2)
    {
Re_insert4_2:
    a2 = (rand() % (8 - 16 + 1)) + 8;
    b2 = (rand() % (16 - 20 + 1)) + 16;

    if (a2>b2 || a2==b2 || b2-a2<2 || b2-a2>8 || (b2-a2)%2!=0)
    {
        goto Re_insert4_2;
    }

    printf("\nStudent %d: Interval orar -> [%d - %d]", student+1, a2, b2);

    if (maxA <= a2)
    {
        maxA = a2;
    }

    if (maxB >= b2)
    {
        maxB = b2;
    }
    if (student == 0)
    {
        student++;
    }

    a1 = a2;
    b1 = b2;
}

if (maxA < maxB)
{
    printf("\nIntervalul orar in care toti studentii sunt prezenti la facultate este: [%d - %d]", maxA, maxB);
}
else
{
    printf("\nNu exista un interval orar in care toti studentii sunt prezenti la facultate .");
}

return 0;
}

```