

1. Convertire

| | | | |
|-----|----|---|---|
| 347 | 43 | 5 | 0 |
| 3 | 3 | 5 | |

$$347_8 = 533$$

| | | | |
|------|-----|----|---|
| 2755 | 172 | 10 | 0 |
| 3 | 12 | 10 | |

$$2755_{16} = AC3$$

| | | | | | |
|----|----|---|---|---|---|
| 20 | 10 | 5 | 2 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | |

$$20_2 = 10100$$

2. Convertire in baza 10

$$1000101 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

$$= 64 + 4 + 1 = 69$$

$$357 = 3 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0$$

$$= 3 \cdot 64 + 40 + 7 = 192 + 47 = 239$$

$$C7A = 12 \cdot 16^2 + 7 \cdot 16^1 + 10 \cdot 16^0$$

$$= 12 \cdot 256 + 112 + 10$$

$$= 3072 + 122 = 3194$$

3. Operatii

a)

$$FA_{16} = 15 \cdot 16^1 + 10 \cdot 16^0$$

$$= 240 + 10 = 250$$

$$23_{16} = 2 \cdot 16^1 + 3 \cdot 16^0$$

$$= 32 + 3 = 35$$

$$250 + 35 = 285$$

| | | | |
|-----|----|---|---|
| 285 | 17 | 1 | 0 |
| 13 | 1 | 1 | |

$$285 = 11D$$

b)

$$1101\ 0010_2 + 1110\ 1101_2 =$$

$$1101\ 0010$$

$$1110\ 1101$$

$$----- +$$

$$1.1011\ 1111$$

c)

$$1176_8 - 723_8 =$$

$$C1(723) = 054_8$$

$$C2(723) = 055_8 = -723_8$$

$$1176_8 + 055_8 = 1253_8$$

4. Convertire din baza 2 in baza 8 si 16

a)

$$(10101010.110010)_2 = (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) + (0 \times 2^{-4}) + (1 \times 2^{-5}) + (0 \times 2^{-6}) = 128 + 32 + 8 + 2 + \frac{1}{2} + \frac{1}{4} + \frac{1}{32} = 170.78125$$

170.78125 \times baza la puterea numarului de cifre dupa virgula

$$170.78125 \times 8^5 = 5596160$$

| | | | | | | | | |
|---------|--------|-------|-------|------|-----|----|---|---|
| 5596160 | 699520 | 87440 | 10930 | 1366 | 170 | 21 | 2 | 0 |
| 0 | 0 | 0 | 2 | 6 | 2 | 5 | 2 | |

$$(10101010.110010)_2 = (252.62)_8$$

$$170.78125 \times 16^5 = 179077120$$

| | | | | | | | |
|-----------|----------|--------|-------|------|-----|----|---|
| 179077120 | 11192320 | 699520 | 43720 | 2732 | 170 | 10 | 0 |
| 0 | 0 | 0 | 8 | 12 | 10 | 10 | |

$$(10101010.110010)_2 = (AA.C8)_{16}$$

b)

$$(1100010.11101)_2 = (1 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3}) + (0 \times 2^{-4}) + (1 \times 2^{-5}) = 64 + 32 + 2 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{32} = 98 + 0.5 + 0.25 + 0.125 + 0.03125 = 98.90625$$

$$98.90625 \times 8^5 = 3240960$$

| | | | | | | | | |
|---------|--------|-------|------|-----|----|----|---|---|
| 3240960 | 405120 | 50640 | 6330 | 791 | 98 | 12 | 1 | 0 |
| 0 | 0 | 0 | 2 | 7 | 2 | 4 | 1 | |

$$(1100010.11101)_2 = (142.72)_8$$

$$98.90625 \times 16^5 = 103710720$$

| | | | | | | | |
|-----------|---------|--------|-------|------|----|---|---|
| 103710720 | 6481920 | 405120 | 25320 | 1582 | 98 | 6 | 0 |
| 0 | 0 | 0 | 8 | 14 | 2 | 6 | |

$$(1100010.11101)_2 = (62.E8)_{16}$$

5. Convertire din baza 10 in baza 4

$$175.1285 \times 4^5 = 179331.584$$

| | | | | | | | | | |
|------------|-------|-------|------|-----|-----|----|----|---|---|
| 179331.584 | 44832 | 11208 | 2802 | 700 | 175 | 43 | 10 | 2 | 0 |
| 4 | 0 | 0 | 2 | 0 | 3 | 3 | 2 | 2 | |

$(175.1285)_{10} = (2233.02004)$
 (nu stiu exact cum sa fac aici)

6. Algoritmi in pseudocod

Sortare prin metoda bulelor

n, aux, numere naturale

a[101], sir de numere naturale

k, variabila de tip intrerupator

se citeste n si un sir de numere a[101] de n elemente

executa

```
{
    k <- 1 ;
    pentru i <- 0 , i < n - 1, i <- i + 1 executa
        daca a[i] > a[i+1]
            {
                k <- 0
                aux <- a[i]
                a[i] <- a[i+1]
                a[i+1] <- aux
            }
        Sfarsit daca
    Sfarsit pentru
}
cat timp ≠ k
se afiseaza sirul a
```

7. Algoritmul lui Euclid (CMMDC)

n, m numere naturale

se citesc n si m

cat timp n ≠ m

daca n > m

n <- n - m;

sfarsit daca

altfel

m <- m - n;

sfarsit altfel

se afiseaza n

8.

Descompunerea unui număr în factori primi

Citim un numar n. Parcurgem cu o structura repetitiva cu test initial (for) de la l = 2 pana la n inclusiv (cu cea de a doua conditie ca n sa fie diferit de 1) si parcugem din 1 in 1. Daca in urma impartirii lui l la n nu avem rest, atunci cu un while se imparte l la n si se numara de cate ori se face aceasta impartire intr-un p, dupa care (in acest daca) se afiseaza i-ul (factorul prim) si p-ul (puterea factorului prim).

Determinarea tuturor numerelor prime până la un n citi

Prima data scriem o functie prim, cu ajutorul careia putem determina daca un numar este prim sau nu, aceasta functie returneaza 0 daca numarul nu este prim si respectiv 1 daca numarul este prim.

In functia prim: daca numarul este mai mic decat 2 atunci se returneaza 0; altfel daca numarul este divizibil cu 2 si este diferit de 2 se returneaza 0; altfel se parcurge cu un for de la $d = 3$ pana la radical din numarul pe care vrem sa il testam daca este prim si parcurgerea este din 2 in 2; daca d se imparte la numarul nostru si nu avem rest, atunci se returneaza 0; in final, daca niciuna dintre conditii nu a fost indeplinita, atunci se returneaza 1, adica numarul nostru este prim.

In programul principal citim un numar n si parcurgem cu un for de la $d = 3$ pana la n inclusiv si din 2 in 2. Daca apelam functia prim de d si ne returneaza 1 inseamna ca d este un numar prim si se afiseaza.

9.

```

numar_cifre(n)
n, k numere naturale
{
    k <- 0
    cat timp n ≠ 0
    {
        K <- k + 1
        n <- n / 10
    }
    returneaza k
}

```

Program principal

```

{
    a, b, cifa, cifb, aux, i, j, nr, t, k = 0 numere naturale
    s[20] = { 0 } sir
    se citesc a, b
    t <- numar_cifre(a) + numar_cifre(b)
    aux <- b
    cat timp a ≠ 0
    {
        cifa <- a % 10
        a <- a / 10
        b <- aux
        k <- k + 1
        i <- k, j <- k + 1
        cat timp b ≠ 0
        {
            cifb <- b % 10
            b <- b / 10
            nr <- cifa * cifb
            s[i] <- s[i] + nr % 10, nr <- nr / 10
        }
    }
}

```

```

        s[j] <- s[j] + nr % 10
        i <- j
        j <- j + 1
    }
    Sfarsit cat timp
}
Sfarsit cat timp
pentru i <- 1, i <= t, l <- l + 1
{
    afiseaza s[i] " "
}
Sfarsit pentru
Afiseaza linie noua
z <- 0 numar natural
pentru i <- 1, i <= t, l <- l + 1
{
    Daca s[i] >= 10
    {
        s[i + 1] <- (s[i] / 10) % 10 + s[i + 1]
        daca s[i + 1] = 0
        {
            z <- z + 1
        }
        Sfarsit daca
        alfel
        {
            z <- 0
        }
        Sfarsit altfel
        s[i] <- s[i] % 10
    }
}
Sfarsit pentru
daca s[i + 1] = 0
{
    z <- z + 1
}
Sfarsit daca
alfel
{
    z <- 0
}
Sfarsit altfel
t <- t - z

```

```

        pentru i <- t, i >= 1, l <- l - 1

        {
            afiseaza s[i]
        }
        Sfarsit pentru
    }

```

Pun algoritmul si in c++ daca vreti sa il testati:

```

#include<iostream>
using namespace std;

```

```

int numar_cifre(int n)

```

```

{
    int k = 0;
    while (n)
    {
        k++;
        n = n / 10;
    }
    return k;
}

```

```

int main()

```

```

{
    int a, b, cifa, cifb, aux, i, j, nr, t, k = 0;
    int s[20] = { 0 };
    cin >> a >> b;
    t = numar_cifre(a) + numar_cifre(b);
    aux = b;
    while (a)
    {
        cifa = a % 10;
        a = a / 10;
        b = aux;
        k++;
        i = k; j = k + 1;
        while (b)
        {
            cifb = b % 10;
            b = b / 10;
            nr = cifa * cifb;
            s[i] = s[i] + nr % 10; nr = nr / 10;
            s[j] = s[j] + nr % 10;

```

```
        i = j;
        j++;
    }
}
for (int i = 1; i <= t; i++)
{
    cout << s[i] << " ";
}
cout << endl;
int z = 0;
for (int i = 1; i <= t; i++)
{
    if (s[i] >= 10)
    {
        s[i + 1] = (s[i] / 10) % 10 + s[i + 1];
        if (s[i + 1] == 0)
        {
            z++;
        }
        else
        {
            z = 0;
        }
        s[i] = s[i] % 10;
    }
}
if (s[i + 1] == 0)
{
    z++;
}
else
{
    z = 0;
}
t = t - z;
for (int i = t; i >= 1; i--)
{
    cout << s[i];
}
}
```