BAZE DE NUMERAȚIE – EVALUARE

prof. dr. Daniela Marcu
prof. Ovidiu Marian Marcu

PARTEA I. Fișe de lucru

Fișa de lucru nr. 1 – Noțiuni fundamentale și conversii

(Nivel: începător - 20 de minute)

Exercițiul 1. Precizează cifrele utilizate în fiecare sistem de numerație:

a) baza 2;

b) baza 8;

c) baza 10;

d) baza 16.

Exercițiul 2. Transformă în baza 10 următoarele numere:

a) $(10101)_2$

b) $(110110)_2$

c) $(237)_8$

d) $(3A)_{16}$

Exercițiul 3. Transformă din baza 10 în baza 2:

a) 19

b) 42

c) 73

Exercițiul 4. Explică de ce, la transformarea din baza 10 într-o bază (b), resturile împărțirilor succesive trebuie citite în ordine inversă.

Fișa de lucru nr. 2 – Algoritmi de conversie

(Nivel: mediu - 40 de minute)

Exercițiul 1. Scrie pseudocodul pentru transformarea unui număr natural din baza 10 în baza 2. Tratează și cazul în care numărul este 0.

Exercițiul 2. Scrie pseudocodul pentru transformarea unui număr binar în baza 10, folosind formula cu puterile lui 2.

Exercițiul 3. Transformă rapid din baza 2 în baza 16, grupând biții câte patru:

a) $(110101101)_2$

b) $(11111111)_2$

c) $(101011110010)_2$

Exercițiul 4. Transformă rapid din baza 8 în baza 2:

a) $(572)_8$

b) $(143)_8$

Exercițiul 5. Se citește un număr n, reprezentat prin cifre în baza b. Descrie un algoritm care determină valoarea numărului în baza 10, fără să utilizeze explicit puterile bazei ☺

Fișa de lucru nr. 3 – Aplicații algoritmice

(Nivel: avansat - 45–50 de minute)

Exercițiul 1. Transformă $(2AF)_{16}$ în baza 10.

Exercițiul 2. Transformă numărul $(2023)_4$ în baza 7.

Exercițiul 3. Folosește înmulțirea à la russe pentru a calcula 27 * 14. Explică, pas cu pas, algoritmul.

Exercițiul 4. Reprezintă numărul 19 în baza Fibonacci.

Exercițiul 5. Un număr binar are 25 de biți. Câte cifre hexazecimale sunt necesare pentru reprezentarea lui? Explică de ce poate fi necesară completarea cu zerouri la stânga.

Exercițiul 6. Scrie un algoritm care citește un număr în baza 10 și o bază (b), 2 <= b <=16, și afișează reprezentarea numărului în baza (b).

PARTEA A II-A. Probleme cu rezolvări în C++

> Enunțurile complete ale problemelor din această secțiune se pot accesa de pe site-ul pbinfo.ro.

Exercițiu: scrie pseudocodul echivalent pentru fiecare problemă propusă în această secțiune.

ID: #429 CifBin

Enunțul pe scurt: Se citește un număr natural n. Să se determine câte cifre 0 și câte cifre 1 are reprezentarea în baza 2 a acestui număr.

Rezolvare în C++

```
#include <iostream>
using namespace std;
int main()
{
    int n,c0=0,c1=0;
    cin>>n;
    while(n)
    {
        if(n%2)
            c1++;
        else
        c0++;
        n=n/2;
    }
    cout<<c0<<" "<<c1;
    return 0;
}
```

Explicații

- `c0` numără cifrele `0` din reprezentarea binară a lui `n`.
- `c1` numără cifrele `1` din reprezentarea binară a lui `n`.
- `n % 2` verifică ultima cifră binară: dacă restul este `1`, incrementăm `c1`, altfel incrementăm `c0`.
- `n = n // 2` elimină ultima cifră binară.

Exemplu:

```
123
1 6
```

ID: #428 transfB

Enunțul pe scurt: Se citește un număr b, reprezentând o bază de numerație și apoi n numere naturale, reprezentând cifrele unui număr scris în baza b. Determinați transformarea acestui număr în baza 10.

Rezolvare în C++

```
#include <iostream>
using namespace std;
int main()
{
    int n,b,x,i,p=1,y=0;
    cin>>b>>n;
```

Explicații

Variabila **p** reține inițial b^{n-1}.

Se citesc cele **n** cifre, iar fiecare cifră **x** este înmulțită cu puterea corespunzătoare a bazei și adăugată la y.

Exemplu:

```
    x=n-1;
    while(x)
    {
      p=p*b;
      x--;
    }
    for(i=1;i<=n;i++)
    {
      cin>>x;
      y=y+x*p;
      p=p/b;
    }
    cout<<y;
    return 0;
}
```

```
5 6
1
0
2
2
4
1
Rezultat:
3446
```

ID: #3348 SumaPuteri2

Enunțul pe scurt: Se dă un număr natural n. Descompuneți numărul n ca sumă de puteri ale lui 2.

Rezolvare în C++

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    long long int sol=1;
    int cn=n,k=-1,v[64];
    while(cn>=1)
    {
        v[++k]=cn%2;
        cn/=2;
    }
    for(int i=0;i<=k;i++)
    {
        if(v[i]==1)
            cout<<sol<<" ";
            sol*=2;
```

Explicații

Explicație:

- v reține cifrele reprezentării binare a lui n, de la dreapta la stânga.

- sol începe de la 1 și se dublează la fiecare pas: 1, 2, 4, 8, 16....

- Dacă cifra binară este 1, se afișează puterea corespunzătoare a lui 2.

Exemplu:

```
122
Rezultat:
2 8 16 32 64
```

```
    }
    return 0;
}
```

ID: #2560 bits

Enunțul pe scurt: Se dă un număr natural N. Determinați valoarea unor anumiți biți din reprezentarea sa în baza 2.

Autor: Marius Niculi - Balcaniada de Informatică 2018, ziua de antrenament

Rezolvare în C++

Varianta 1:

```
#include<bits/stdc++.h>
using namespace std;
ifstream f("bits.in");
ofstream g("bits.out");
short int a[70],m;
unsigned long long n;
int main()
{
    int i,Q;
    f>>n>>Q;
    while(n)
    {
        a[m++]=n%2;
        n/=2;
    }
    while(Q--)
    {
        f>>i;
        g<<a[i];
    }
    return 0;
}
```

Varianta 2:

```
#include<bits/stdc++.h>
using namespace std;
unsigned long long n, b, i, Q;
int main ()
{
```

Explicații

Varianta 1 determină valoarea unor biți din reprezentarea binară a numărului n. Biții sunt calculați și memorați într-un vector, iar apoi sunt afișați la cerere.

Varianta 2 folosește operații pe biți.

Bucla **for** se execută de Q ori.

La fiecare repetare:

1. Se citește poziția b din fișier.
2. Se calculează valoarea bitului de pe poziția respectivă.
3. Se scrie rezultatul în fișierul de ieșire.

Cum funcționează `(n>>b)&1)?`

Scopul este să aflăm valoarea bitului de pe poziția b din numărul n, numerotând biții de la 0, începând din dreapta.

Exemplu: n = 13, b = 2

Numărul 13 în baza 2 este 1101.

1. Scriem numărul în binar

1 - poziția 3 (2^3)

1 - poziția 2

0 - poziția 1

1 - poziția 0

Bitul de pe poziția 2 este 1.

2. Aplicăm deplasarea pe biți (shift right): n >> b

13 >> 2 înseamnă că deplasăm biții cu două poziții spre dreapta.

1101 → 0011

```
    ifstream fin ("bits.in");
    ofstream fout("bits.out");
    fin>>n>>Q;
    for (i=1;i<=Q;i++)
    {
        fin>>b;
        fout<<((n>>b)&1);
    }
    return 0;
}
```

Rezultatul este 3 în baza 10.

3. Aplicăm & 1

Comparăm ultimul bit al rezultatului cu 1:

0011

&

0001

0001

= 1

Rezultatul final este 1.

De ce folosim & 1?

Operatorul & compară fiecare pereche de biți. Rezultatul este 1 numai când ambii biți sunt 1.

Bitul din număr	& 1	Rezultat
0	1	0
1	1	1

Astfel, & 1 păstrează doar ultimul bit și elimină toți ceilalți biți din rezultat.

Rezultat: 1

13 deplasat cu 2 poziții la dreapta dă 3, iar ultimul bit al acestuia este 1.

Astfel aflăm direct valoarea bitului de pe poziția b.

Probleme propuse (de pe site-ul pbinfo.ro):

Ușoare	#122, #427, #2318
Medii	#3349, #945, #982
Concurs/Dificile	#946, #1300, #3957