

Exemplul 1: Hello world!

```
#include<iostream>
using namespace std;
int main(){
    cout << "Hello world!" << endl;
    return 0;
}
```

Baze de numerație

- numărarea într-o bază oarecare
- trecerea dintr-o bază în alta
- adunarea într-o bază dată
- binar<-> hexal

Exemplul 2: Exercițiul final din Tema 01:

$$s_n = 2^{2^1} + 2^{2^2} + 2^{2^3} + \dots + 2^{2^n}$$

```
#include<iostream>
using namespace std;
const int dim = 500;
typedef int registru[dim];

void initializeaza(registru w){           //w=1
    w[0] = 1;
    for (int i = 1; i<dim; i++) w[i] = 0;
    return;
}
void dubleaza(registru w){
    int t = 0;                          // transportul initial e nul
    for (int i = 0; i<dim; i++){
        int aux = t + 2 * w[i];
        w[i] = aux % 10; //noul w[i]=ultima cifra a lui aux
        t = aux / 10;    //noul t=prima cifra a lui aux
    }
    return;
}
void afiseaza(registru w){
    int i;
    for (i = dim - 1; i >= 0 && w[i] == 0; i--)
        ; //sarim peste zerourile din fata
    if (i<0) cout << 0;
    else for (; i >= 0; i--){
        cout << w[i]; //afisam incepand cu prima cifra nenula
        if (i % 3 == 0) cout << ' ';
    }
    cout << "*" << endl;
    return;
}
```

```

void cumuleaza(registru a, registru b){
    int t = 0;           // transportul initial e nul
    for (int i = 0; i<dim; i++){
        int aux = t + a[i] +b[i];
        a[i] = aux % 10;
        t = aux / 10;
    }
    return;
}
int main(){
    int n = 10;
    registru s, t;
    initializeaza(s);
    initializeaza(t);
    s[0] = 0;
    int p = 1, k = 1;
    for (int i = 0; i <= n; i++){
        for (; k <= p; k++){
            dubleaza(t);
        }
        cumuleaza(s, t);
        afiseaza(s);
        p *= 2;
    }
    return 0;
}

```

Prefixe utilizate în informatică

- bit (b), byte=octet (B=8b), word=cuvânt 2B=16b, double word=dword=32b
- kilo=K=2¹⁰=1024;
- mega=M=K², giga=G=K³, tera=T=K⁴, peta=P=K⁵, exa=E=K⁶, zetta=Z=K⁷, yotta=Y=K⁸

Generalitati despre structura hardware a unui PC

- placa de bază (motherboard)
- procesor
- memoria RAM (spațiu de lucru)
- memoria BIOS (softul minimal)
- chipseturi - magistrala de date (bus-ul)
- periferice: dispozitive IO, dispozitive de stocare (FD, HDD, CD, etc), porturi
- plăci de extensie (card-uri)- placa video, placa de rețea, placa de sunet, tuner tv, etc

Generalități despre dotarea software a unui PC

- cum funcționează procesorul IA-32:
 - registrii, cod mașină (limbaj de asamblare), **debugger.doc**
- organizarea memoriei – **stack/heap**
- sistem de operare – managerul de programe
- etapele realizării unui program:
 - editarea fișierelor sursă (limbaj de programare)
 - compilarea sau interpretarea, rezultă module în cod mașină
 - link-edit-area (editarea de legături – building)
 - este construit programul cu adrese relative

- depanarea (debugging) – rezultă cod supravegheat
- executarea
 - lansarea codului executabil de către sistemul de operare
 - programul primește adrese absolute de memorie

Generalități despre limbaje de programare

-istoric:

- 1950-59 FORTRAN, COBOL, ALGOL, LISP
- 1960-69 BASIC, B, LOGO
- 1970-79 Pascal, C
- 1980-89 C++, Object Pascal-Delphi, Visual Basic
- 1990-99 Java, Haskell
- 2000-10 C#

-paradigme:

- programare imperativă
 - procedurală - Basic, C, Pascal
 - orientată obiect (POO) - C++, Java, C#, Delphi, Visual Basic
- programare funcțională - Lisp, Haskell

Evoluția limbajului C

- apariție: 1970 - Dennis M. Ritchie, AT&T Bell Laboratories – sistemul de operare UNIX
- manual: 1978 - Brian W. Kernighan, Dennis M. Ritchie, *The C programming language*.
- C++ 1986 - Bjarne Stroustrup, *The C++ programming language*