



Programarea calculatoarelor

Profesor: Maria Guțu

Declararea constantelor

În standardul **K&R C** inițializarea constantelor are următoarea sintaxă:

#define **name** **value**, plasată până la funcția **main()**.

#define - directiva compilatorului,

name – numele constantei,

value – valoarea constantei.

Exemple: **#define** **K** **35**

#define **salariu** **700**

#define **PI** **3.14**

Declararea constantelor

Standardul ANSI-C permit inițializarea constantelor în 2 moduri:

1. folosirea directivei **#define**;
2. folosirea cuvântului rezervat **const**.

Sintaxa generală:

const **type name** = **value**, plasată după funcția **main()**.

const - cuvântul rezervat pentru inițializarea constantelor;

type name - cuvântul rezervat pentru tipul de date și numele variabilei;

value – valoarea constantei.

Declararea constantelor

Exemplu:

```
main() {  
  const float PI = 3.14;  
  const int K = 35;  
  ... ;  
  const float salariu = 3250.25;  
  ... ;  
}
```

Instrucțiuni

Instrucțiunile pot fi împărțite în două grupe:

- 1. Instrucțiunile perioadei de compilare**, caracterizează datele din program;
- 2. Instrucțiunile perioadei de îndeplinire a programului**, determină acțiuni de prelucrare a datelor în conformitate cu algoritmul dat.

Tipurile instrucțiunilor

Instrucțiuni pot fi **simple**, **compuse** și **blocuri**.

Simplă, instrucțiunea care nu conține în componența sa o altă instrucțiune.

Instrucțiuni simple: **expresii**, instrucțiunea continuării **continue**, instrucțiunea de terminare **break**, instrucțiunea de întoarcere **return**, instrucțiunea de salt necondiționat **goto**.

Tipurile instrucțiunilor

Instrucțiuni pot fi **simple**, **compuse** și **blocuri**.

Compusă, instrucțiunea ce conține în componența sa alte instrucțiuni.

Instrucțiuni compuse: instrucțiunea condițională **if – else**, instrucțiunile ciclului **for, while, do while** și instrucțiunea de selecție **switch**.

Tipurile instrucțiunilor

Instrucțiuni pot fi **simple**, **compuse** și **blocuri**.

Bloc – un șir de instrucțiuni luate în acolade(**{}**). **Instrucțiunile** blocului **se îndeplinesc consecutiv** în ordinea înscrierii lor în interiorul blocului. La începutul blocului pot fi descrise **variabilele interioare**. În așa caz se spune, că aceste variabile sunt localizate în interiorul blocului, există și lucrează numai în interiorul blocului și se pierd în afara lui.

Etichete

Orice instrucțiune poate fi marcată de un identificator, numit **etichetă**.

Eticheta este separată de instrucțiune prin două puncte.

Sintaxa generală:

Etichetă : **corpul_instrucțiunii**;

Eticheta se folosește doar în cazul când se folosește saltul necondiționat cu ajutorul instrucțiunii **goto**.

Etichete

Exemplu:

```
main() {  
    goto et;  
    printf("Acest mesaj nu se va afisa\n");  
    et: printf(" Acest mesaj se va afisa \n");  
    return 0;  
}
```

Instrucțiunea de salt necondiționat GOTO

Instrucțiunea **goto** transmite controlului execuției programului la o instrucțiune marcată cu o etichetă.

Sintaxa generală:

goto_etichetă;

Instrucțiunea de salt necondiționat GOTO

Exemplu:

```
#define lim 5
main() {
    int n = 1; float y = 0;
    m1: ++n;
    if (n <= lim) { y += (float)n / (n + 5); goto m1;}
}
```

Instrucțiuni expresie

Orice expresie va deveni instrucțiune dacă se va termina cu (;). Deci, instrucțiunea-expresie are următoarea formă:
expresie_;

Exemplu:

```
int x, y, z;
```

```
x = -3 + 4 * 5 - 6;
```

```
y = 3 + 4 % 5 - 6;
```

```
z = -3 * 4 % -6 / 5;
```

Instrucțiuni de ramificare (condiționale)

Instrucțiunea de ramificare IF și IF-ELSE

Instrucțiunea **if** este o instrucțiune compusă și sintaxa ei admite unul din următoarele formate:

1. **if**_(**expresie**)_**instrucțiune**;
2. **if**_(**expresie**)_**instrucțiune1**_else_**instrucțiune2**;

Exemplu:

```
if (d= =c) { ++d; c += 2;
```

```
if (a>b) max=a; else max=b;
```

Instrucțiunea de selectare SWITCH

Instrucțiunea de selectare e destinată pentru selectarea unei singure variante dintr-o mulțime posibilă de ramificări în program.

Sintaxa generală:

```
switch (expresie) {  
    case expr_const1: instrucțiune1;  
    case expr_const2: instrucțiune2; ...  
    case expr_const_n: instrucțiune_n;  
default: instrucțiune; }
```

NB: Valoarea expresiei trebuie să fie de tip **int** sau **char**.

Instrucțiunea de selectare SWITCH

Exemplu:

```
int main() {  
    char c;  
    printf("Alegeți o opțiune:\n\t[a] afișare\n\t[s] ștergere\n\t[e] exit\n");  
    scanf("%c", &c);  
    printf("Ați ales: ");  
    switch (c) {  
        case 'a': printf("afișare"); break;  
        case 's': printf("ștergere"); break;  
        case 'e': printf("ieșire"); break;  
        default: printf("O opțiune inexistentă"); break;  
    } return 0; }
```


Instrucțiunea de întrerupere BREAK

Instrucțiunea `break`, pe lângă utilizarea descrisă la instrucțiunea `switch`, poate fi folosită pentru a ieși forțat dintr-o instrucțiune de repetiție.

Exemplu:

```
int main () { //suma a 5 numere pozitive
    int i; double number, sum = 0.0;
    for (i = 1; i <= 5; ++i) {
        printf("Enter a n%d: ", i);
        scanf("%lf", &number);
        if (number <= 0.0) { break; }
        sum += number; // sum = sum + number;
    }
    printf("Sum = %.2lf", sum);
    return 0; }
```