

Laborator 3-4. Preprocesarea în limbajul C: directivele `#include`, `#define`. Funcții de intrare: `getch()`, `getche()`, `gets()`, `getchar()`, `scanf()`. Funcții de ieșire: `putch()`, `puts()`, `putchar()`, `printf()`

3.1. Preprocesarea în limbajul C

Preprocesarea realizează prelucrări de macrosubstituție (definirea de constante simbolice sau de macrocomenzi de tip funcție), compilare condiționată și includere de fișiere, înainte de compilare. Instrucțiunile sau directivele de preprocesare sunt precedate de simbolul `#` și de obicei sunt amplasate la începutul programului.

3.1.1 Directiva `#include`

Directiva `#include` permite includerea la compilare a anumitor fișiere, tratând textul din fișierul specificat, ca și cum ar apărea în fișierul curent.

Sintaxă:

```
#include "numefisier"  
#include <numefisier>
```

În prima formă, se caută fișierul `numefisier` în directorul curent, apoi în directorul `include`. În cea de a doua formă, nu se caută în directorul curent.

Exemple:

```
#include <stdio.h>  
#include "functie.cpp"
```

3.1.2 Directiva `#define`

Directiva `#define` definește o macrosubstituție (înlocuire, cu sau fără un set de parametri formali, de tip funcție).

Sintaxă:

```
#define id1[(id2,...)] sir_de_caractere
```

Identificatorii nu pot conține spații, dar `sir_de_caractere` poate conține orice caracter, cuvânt sau propoziție, conținând oricâte spații. Toate aparițiile identificatorului `id1` în program vor fi înlocuite cu șirul de caractere. Dacă `id1` este urmat de o paranteză deschisă, identificatorii care apar după paranteză sunt tratați ca și parametri ai unei funcții.

Exemple:

```
#define PI 3.14  
#define FILL '*'  
#define ARIE_CERC(x) PI*x*x
```

3.1.3 Exemple de programe

Programul `ex_defi.cpp`. Definește ca și constante simbolice constanta `PI`, constanta `RAZA`. Calculează circumferința și aria cercului cu raza `RAZA`.

```
/*definirea de constante simbolice*/  
#include <stdio.h>  
#define PI 3.14  
#define RAZA 7.0  
void main(void)  
{  
printf("Circumferinta cercului cu raza %f este: %f\n", RAZA, 2*PI*RAZA);  
printf("Aria cercului cu raza %f este: %f\n", RAZA, PI*RAZA*RAZA);  
}
```

```
}
```

Programul *ex_macro.cpp*. Definește macrocomanda de tip funcție, având numele *ABS*, parametrul de intrare *x* și calculează modulul sau valoarea absolută din *x*, folosind operatorul condițional ternar.

```
/*definirea de macrocomenzi tip functie*/  
#include <stdio.h>  
#define ABS(x) (x)<0?- (x):(x)  
void main(void)  
{  
printf("Valoarea absoluta din -10 si 10 este: %d %d\n", ABS(-10), ABS(10));  
}
```

3.2 Funcții de intrare

3.2.1 Funcția *getch()*

Sintaxă:

```
int getch(void);
```

Prototipul funcției este definit în *conio.h*.

Citește un caracter de la *stdin*, fără ecou pe ecran. Caracterul este disponibil imediat.

3.2.2 Funcția *getche()*

Sintaxă:

```
int getche(void);
```

Prototipul funcției este definit în *conio.h*.

Citește un caracter de la *stdin*, cu ecou pe ecran. Caracterul este disponibil imediat.

3.2.3 Funcția *gets()*

Sintaxă:

```
char *gets(char *string);
```

Prototipul funcției este definit în *stdio.h*.

Citește un șir de caractere de la *stdin*. Returnează un pointer la șirul de caractere argument.

3.2.4 Funcția *getchar()*

Sintaxă:

```
int getchar(void);
```

Prototipul funcției este definit în *stdio.h*.

Citește un caracter de la tastatură. Returnează caracterul citit, după ce a fost convertit la un întreg fără semn.

3.2.5 Funcția *scanf()*

Sintaxă:

```
int scanf(const char *format [, argument, ...]);
```

Prototipul funcției este definit în *stdio.h*.

Produce o intrare cu format de la *stdin*. Returnează un număr întreg, reprezentând numărul de valori citite corect.

3.3 Funcții de ieșire

3.2.1 Funcția *putch()*

Sintaxă:

```
int putch(int ch);
```

Prototipul funcției este definit în *conio.h*.

Produce ieșirea unui caracter în fereastra de text. Returnează caracterul afișat, sau în caz de eroare, valoarea EOF. Utilizează atributele de afișare și culoare curentă.

3.2.2 Funcția *puts()*

Sintaxă:

```
int *puts(const char *);
```

Prototipul funcției este definit în *stdio.h*.

Trimite un șir de caractere la *stdout* și face trecerea la rând nou. Returnează ultimul caracter scris sau EOF în caz de eșec.

3.2.3 Funcția *putchar()*

Sintaxă:

```
int getchar(int c);
```

Prototipul funcției este definit în *stdio.h*.

Trimite un caracter spre ecran. Returnează caracterul *c*, iar în caz de eroare, valoarea EOF.

3.2.4 Funcția *printf()*

Sintaxă:

```
int printf(const char *format [, argument, ...]);
```

Prototipul funcției este definit în *stdio.h*.

Produce o ieșire cu format în *stdout*. Returnează un număr întreg, reprezentând numărul de caractere afișate, sau în caz de eroare, valoarea EOF.

3.2.5. Exemple

Ex1. Să se tipărească pe ecran literele 'a', 'b', 'c', la coordonatele: (10,5), (20,10), (30,15).

```
#include <conio.h>
void main(void)
{
    textattr( 23 );
    clrscr();
    gotoxy( 10, 5 );
    putchar( 'a' );
    gotoxy( 20, 10 );
    putchar( 'b' );
    gotoxy( 30, 15 );
    putchar( 'c' );
    getch();
}
```

Ex2. Să se citească cu ecou un caracter de la terminal și apoi să se afișeze pe ecran urmat de punct.

```
#include <conio.h>
void main(void)
{
    int caract;
    caract=getche();
    putchar(caract);
    putchar('.');
    getch();
}
```

Ex3. Să se introducă de la tastatură pe rând trei caractere, după care să se tipărească acestea pe același rând.

```
#include <stdio.h>
#include <conio.h>
void main (void)
{
    int caract1, caract2, caract3;
    printf(" \nIntroduceti primul caracter:");
    caract1=getche();
    printf(" \nIntroduceti al doilea caracter:");
    caract2=getch();
    putchar(caract2);
    printf(" \nIntroduceti al treilea caracter:");
    caract3=getche();
    printf("\nAti introdus caracterele:");
    putchar(caract1);
    putchar(caract2);
    putchar(caract3);
    printf("\nApasati orice tasta pentru continuare");
    getch();
}
```

Ex4. Să se citească de la tastatură litere și să se afișeze cu majuscule dacă sunt scrise normal și invers.

```
#include <stdio.h>
#include <conio.h>
void main (void)
{
    int litera;
    printf(" \nIntroduceti litera:");
    litera=getche();
    if (litera<=122 &&litera>=97)
        putchar(litera-'a'+'A');
    else
        if( litera<=90 &&litera>=65)
            putchar(litera+'a'-'A');
        else
            printf("\nCaracterul introdus nu e litera");
    printf("\nApasati orice tasta pentru continuare");
    getch();
}
```

Ex5. Să se citească un șir de caractere din tabloul sir și să se afișeze șirul și lungimea acestuia. Pentru operațiile de intrare-ieșire se vor utiliza funcțiile *gets()* și *puts()*.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main(void)
{
    char sir[50];
    printf("\nIntroduceti sirul:");
    gets(sir);
    printf("\nSirul este:");
    puts(sir);
    printf("Lungimea sirului este:%d\n",strlen(sir));
    printf("Actionati orice tasta pentru continuare\n"),
    getch();
}
```

Ex6. Să se scrie un program care citește de la intrarea standard numele și prenumele unei persoane și apoi afișează inițialele acesteia pe un rând, fiecare inițială fiind urmată de un punct.


```

/*Utilizarea functiilor gets() si puts()*/
#include<stdio.h>
#include<conio.h>
void main(void)
{
    char nume[200],prenume[200];
    puts("Introduceti numele persoanei:");
    gets(nume);
    puts("Introduceti prenumele persoanei:");
    gets(prenume);
    puts("Initialele persoanei sunt:");
    putchar(nume[0]);
    putchar('.');
    putchar(prenume[0]);
    putchar('.');
    puts("\nApasati orice tasta pentru continuare");
    getch();
}

```

Ex7. Să se citească de la tastatura litere și să se afișeze cu majuscule dacă sunt scrise normal și invers. Programul se va realiza utilizând funcțiile `getchar()` și `putchar()`.

```

/*Utilizarea functiilor getchar() si putchar()*/
#include <stdio.h>
#include <conio.h>

void main (void)
{
    int litera;
    printf(" \nIntroduceti litera:");
    litera=getchar();
    if (litera<=122 &&litera>=97)
        putchar(litera-'a'+'A');
    else
        if( litera<=90 &&litera>=65)
            putchar(litera+'a'-'A');
        else
            printf("\nCaracterul introdus nu e litera");
    printf("\nApasati orice tasta pentru continuare");
    getch();
}

```

Ex8. Să se citească un caracter imprimabil și să se afișeze:

- precedat și urmat de apostrof într-un câmp de 3 caractere, aliniat la dreapta;
- precedat și urmat de apostrof într-un câmp de 3 caractere, aliniat la stânga.

```

/*Afisarea de caractere imprimabile*/
#include <stdio.h>
#include <conio.h>
void main (void)
{
    int caract1;
    printf(" \nIntroduceti caracterul:");
    caract1=getche();
    printf(" \nAfisare caracter aliniat la dreapta: '%3c'\n",caract1);
    printf("Afisare caracter aliniat la stanga: '%-3c'\n",caract1);
    printf("Apasati orice tasta pentru continuare");
    getch();
}

```

Ex9. Să se afișeze textul "Programarea în limbajul C" folosind specificatori de format.

```

/*Afisarea sirurilor de caractere*/
#include <stdio.h>
#include <conio.h>
void main (void)
{
    char sir[]="Programarea in limbajul C";
    printf("\n!%s!",sir);
    printf("\n!%30s!",sir);
    printf("\n!%-30s!",sir);
    printf("\n!%15s!",sir);
    printf("\n!%15.10s!",sir);
    printf("\n!%-15.10s!",sir);
    printf("\n!%10.2ls!",sir);
    printf("\nApasati orice tasta pentru continuare");
    getch();
}

```

Ex10. Să se afișeze constanta 18648 cu diferiți specificatori de format: %d, %8d, %-8d, %08d, %o, %x, %X.

```

/*Afisarea unei constante cu diferiti specificatori de format*/
#include <stdio.h>
#include <conio.h>
#define C 18648
void main (void)
{
    clrscr();
    printf("\nAfisare in zecimal: !%d!",C);
    printf("\nAfisare in zecimal: !%8d!",C);
    printf("\nAfisare in zecimal: !%-8d!",C);
    printf("\nAfisare in zecimal: !%08d!",C);
    printf("\nAfisare in octal: !%o!",C);
    printf("\nAfisare in hexazecimal: !%x!",C);
    printf("\nAfisare in HEXAZECIMAL: !%X!",C);
    printf("\nApasati orice tasta pentru continuare");
    getch();
}

```

Ex11. Să se citească un șir de caractere din tabloul sir și să se afișeze șirul și lungimea acestuia. Pentru operațiile de intrare/ieșire se vor utiliza funcțiile scanf() și printf().

```

/* citirea unui sir si afisarea sirului si a lungimii acestuia*/
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main(void)
{
    char sir[50];
    printf("Introduceti sirul:");
    scanf("%s", sir);
    printf("\nSirul este: %s", sir);
    printf("\nLungimea sirului este: %d\n", strlen(sir));
    printf("Actionati orice tasta pentru continuare\n");
    getch();
}

```

Ex12. Să se citească în mod conversațional elementele unui șir de numere reale și apoi să se tipărească șirul.

```

/*Citirea si tiparirea elementelor unui sir*/
#include<stdio.h>
#include<conio.h>
#define MAX 50
void main(void)
{
    int n,i;
    double a[MAX];
    printf ("Introduceti dimensiunea sirului n<50:");
    scanf ("%d",&n);
    while(n<0||n>MAX)
    {
        printf("Dimensiune eronata>%d\n",n);
        printf("Introduceti alt n:");
    }
}

```

```

scanf("%d",&n);
}
/*se citesc elementele sirului*/
for (i=0;i<n;i++){
printf("Introduceti elem a(%d) al sirului:",i);
scanf("%lf",&a[i]);
}
/*se tiparesc elementele sirului*/
printf("Elementele sirului sunt:\n");
for(i=0;i<n;i++)
    printf("a(%d)=%lf\n",i,a[i]);
printf("Apasati orice tasta pentru continuare\n");
getch();
}

```

Ex13. Să se citească în mod conversațional elementele unei matrice pătratică având elemente numere reale și apoi să se tipărească matricea sub formă de tablou.

```

/*Citirea si afisarea elementelor unei matrice patratice*/
#include<stdio.h>
#include<conio.h>
#define MAX 30
void main(void)
{
    int i,j,n;
    double a[MAX][MAX];
    printf ("\nIntroduceti dimensiunea matricei a (0 <n <30),n="),
    scanf("%d",&n);
    /*se citesc elementele matricei*/
    for(i=0;i<n;i++){
        for (j=0;j<n;j++) {
            printf("Introduceti elementele a(%d,%d) al lui a :",i,j);
            scanf("%lf",&a[i][j]);
        }
    }
    /*se citesc elementele matricei*/
    printf("Elementele matricei a:\n");
    for(i=0;i<n;i++){
        for (j=0;j<n;j++) {
            printf("%15.6lf",a[i][j]);
        }
        printf("\n");
    }
    printf("Apasati orice tasta pentru continuare\n");
    getch();
}

```

- 3.1. Se vor executa programele date ca exemplu în lucrare și se vor analiza rezultatele obținute.
- 3.2. Scrieți un program care afișează caracterele corespunzătoare codurilor ASCII din intervalul [32,126].
- 3.3. Scrieți un program care citește un șir de litere mici și le afișează sub formă de litere mari.
- 3.4. Scrieți un program care citește un șir de litere mari și le afișează sub formă de litere mici.
- 3.5. Scrieți un program care realizează suma, diferența, produsul și împărțirea a două numere reale. Afișarea se va face sub formă tabelară:

x	y	x+y	x-y	x*y	x/y

- 3.6. Scrieți un program pentru a verifica modul de afișare a valorii lui $\pi = 3.14159265$ cu diferiți descriptori de format.
- 3.7. Scrieți un program pentru afișarea unui întreg citit de la tastatură în octal și hexazecimal.