

ȘIRURI DE CARACTERE

1. SCOPUL LUCRĂRII

În această lucrare se prezintă aplicații cu șiruri de caractere (stringuri).

2. BREVIAR TEORETIC

Stringurile sunt vectori de caractere. Pentru a indica sfârșitul șirului, limbajul C folosește drept caracter final al șirului, caracterul nul (octet de valoare 0). Acesta se poate reprezenta și prin secvența escape ‘\0’.

O variabilă de tipul string cel mai adesea se declară ca vector de caractere. Exemplu:

```
char s[4];
```

În acest caz dimensiunea vectorului trebuie să fie mai mare cu cel puțin o unitate decât lungimea maximă a șirurilor ce urmează a fi memorate în variabila string respectivă.

În cazul variabilei *s* din exemplul dat, se pot memora șiruri de maxim 3 caractere.

Variabila șir poate fi inițializată încă din faza de declarare ca în exemplul următor:

```
char s[4]="abc";
```

Există o serie de funcții ce realizează operații asupra șirurilor de caractere.

Pentru citirea stringurilor de la tastatură se poate folosi funcția *scanf* cu specificatorul de format *%s*, sau funcția *gets*. Cu funcția *scanf*, spre deosebire de funcția *gets*, nu se pot citi șiruri ce conțin în interiorul lor caracterul blank. (La funcția *scanf* caracterele blank, TAB și ENTER sunt separatori între câmpurile de citire).

Biblioteca standard furnizează un număr de funcții de procesare stringuri, ce au prototipurile declarate în fișierul header *string.h*. Dintre acestea, cele mai utilizate sunt:

- funcția *strcpy* ce are prototipul:

```
char *strcpy(char *p, char *const q);
```

copiază șirul *q* în șirul *p*. Funcția returnează un pointer spre copia nou creată.

- funcția *strlen* ce are prototipul:

```
int strlen(const char *s);
```

returnează numărul de caractere din șirul *s* (lungimea șirului). În lungimea șirului nu este inclus și caracterul terminal ‘\0’. Astfel șirul

“abcd” are lungimea, returnată de funcția `strlen`, de 4 caractere și nu de 5.

-funcția `strcmp` ce are prototipul:

*`int strcmp(const char *p, const char *q);`*

compară stringurile `p` și `q` din punct de vedere al ordinei lexicografice. Funcția `strcmp` returnează valoarea 0 - dacă cele 2 stringuri `p` și `q`, sunt egale, returnează un număr negativ - dacă primul șir precede lexicografic (este mai mic) decât cel de-al doilea, sau returnează un număr pozitiv dacă al doilea șir precede pe primul.

3. DESFĂȘURAREA LUCRĂRII

Se vor edita și apoi executa programele descrise în continuare.

Programul nr. 1

Se citește de la tastatură un șir de caractere. Să se afișeze câte cifre conține șirul.

Pentru a testa dacă un caracter ASCII este cifră, se folosește funcția de bibliotecă **`isdigit()`**. Această funcție are prototipul:

`int isdigit(int c);`

Ea returnează 0, dacă argumentul ei (variabila `c`) nu este codul ASCII al unei cifre. Dacă este cod de cifră, returnează un întreg nenul.

Sursa programului:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <ctype.h> //fisier in care este declarata functia
                  //isdigit()

void main(void)
{
    char s[81]; //sir de lungime maxima 80 de caractere
    clrscr();
    printf("Tastați un sir: \n");
    scanf("%s",s);
    int L; //lungimea lui s
    int i;
    int nCif; //numarul de cifre din s
    L= strlen(s);
    nCif=0;
    for(i=0; i<L; i++)
```

```

    if(isdigit(s[i]))nCif++;
    printf("Șirul %s are %d cifre.",s,nCif(s));
    getch();
}

```

Programul nr. 2

Se citește de la tastatură un șir de caractere. Să se afișeze dacă este palindrom (simetric fata de mijlocul lui).

Sursa programului:

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
void main(void)
{
    char s[81]; //șir de lungime maxima 80 de caractere
    clrscr();
    printf("Tastați un șir: \n");
    scanf("%s",s);
    int L=strlen(s); //lungimea lui s
    int i;
    int esteSimetric=1; //presupunem ca este
    for(i=0;i<L/2;i++)
        if(s[i]!=s[L-i-1]){
            esteSimetric=0;
            break;}
    if(esteSimetric)printf("Este palindrom");
    else printf("Nu este palindrom");
    getch();
}

```

Programul nr. 3

Se citește de la tastatură un șir de caractere. Să se afișeze dacă șirul are toate caracterele diferite între ele sau nu.

Sursa programului:

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void main()
{
    char sir[100];

```

```
printf("sir=");
scanf("%s",sir);
int L=strlen(sir);
int i,j;
for(i=0;i<L-1;i++)
for(j=i+1;j<L;j++)
if(sir[i]==sir[j]){
printf("NU sunt diferite.");
getch();
exit(0);}
printf("DA., are toate caracterele diferite.");
getch();
}
```

Programul nr. 4

Se citește de la tastatură un șir de caractere s1. Să se construiască șirul s2, obținut prin inversarea șirului s1.

Exemplu: dacă s1="mar" atunci s2="ram".

Sursa programului:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main(void)
{
clrscr();
char s1[80];
printf("Tastați un sir: \n");
scanf("%s",s1);
char s2[80];
int L; //lungime sir s1
L=strlen(s1);
for(int i=L-1;i>=0;i--)
s2[L-1-i]=s1[i];
//terminatorul de string:
s2[L]='\0';
printf("Sirul inversat este:\n%s",s2);
getch();
}
```

Programul nr. 5

Se citesc de la tastatură, sub formă de șiruri de caractere, două numere naturale foarte mari. Să se afișeze care număr este mai mare.

Sursa programului:

```
#include <stdio.h>
#include <string.h>
void main()
{
    char nr1[100];
    char nr2[100];
    printf("nr1=");
    scanf("%s",nr1);
    printf("nr2=");
    scanf("%s",nr2);
    int L1=strlen(nr1);
    int L2=strlen(nr2);
    if(L1>L2){
        printf("nr1 > nr2");
        return;}
    if(L2>L1){
        printf("nr2 > nr1");
        return;}
    //Sirurile au aceeasi lungime, comparam cifrele corespunzatoare
    // intre ele. Incepem cu cea mai semnificativa cifra (pozitia i=0):
    for(int i=0;i<L1;i++){
        int cifra_nr1=nr1[i]-'0';
        int cifra_nr2=nr2[i]-'0';
        if(cifra_nr1>cifra_nr2){
            printf("nr1 > nr2");
            return;}
        if(cifra_nr2>cifra_nr1){
            printf("nr2 > nr1");
            return;}
    }
    printf("nr1 = nr2"); //Singura posibilitate ramasa:
}
```

Programul nr. 6

Se citește un șir s1. Să se construiască șirul s2 ce conține, în aceeași ordine, cifrele ce apar în s1.

Exemplu: dacă s1="a1b3" atunci s2="13".

Sursa programului:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
void main(void)
{
    char s1[81]; //sirul citit.
    char s2[81]; //sirul ce va contine doar cifrele
    int i;
    clrscr();
    printf("sir=");
    scanf("%s", s1);
    int j=0; //index in sirul s2.
    for(i=0; i<strlen(s1); i++)
        if(isDigit(s1[i])){
            s2[j]=s1[i];
            j++;
        }
    //terminatorul de sir, pentru s2:
    s2[j]=0;
    //afisare sirul s2 construit:
    if(j>0) printf("%s", s2);
    else printf("Nu exista cifre in sirul s1 !");
    getch();
}
```

Programul nr. 7

Se citesc de la tastatură, în trei variabile șir de caractere, numele unui director (folder), numele unui fișier (fără extensie) și extensia lui. Să se construiască într-o altă variabilă șir, numele complet al fișierului.

Sursa programului:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char numeF[41];
    char numeDir[41];
    char extensie[5];
```

```

char numeComplet[100];
clrscr();
printf("nume director: ");
scanf("%s",numeDir);
printf("nume fisier (fara extensie): ");
scanf("%s",numeF);
printf("extensia numelui fisierului: ");
scanf("%s",extensie);
//copiez numele directorului, in stringul ce va contine numele
//complet:
strcpy(numeComplet,numeDir);
//adaugam la acest nume, caracterul \
strcat(numeComplet,"\\");
//adaugam si numele propriu-zis al fisierului:
strcat(numeComplet,numeF);
//adaugam si caracterul .
strcat(numeComplet,".");
//adaugam extensia:
strcat(numeComplet,extensie);
//afisam rezultatul:
printf("%s",numeComplet);
getch();
}

```

Programul nr. 8

Se citesc N cuvinte de la tastatură, într-un vector de stringuri. Să se sorteze în ordine alfabetică acest vector.

Sursa programului:

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
typedef char string80[81];
#define N 5 //numarul de cuvinte
void main()
{
    string80 A[N]; //vectorul de cuvinte
    int i;
    string80 aux; //sir tampon, folosit la sortare
    int suntInv; //semafor, folosit la sortare
    clrscr();

```

```

for(i=0;i<N;i++){
    printf("cuvant=");
    scanf("%s",A[i]);}
//sortare cu metoda bubble sort:
int ultim=N-1;
for(;;){
    suntInv=0;
    for(i=0;i<ultim;i++)
        if(strcmp(A[i],A[i+1])>0){
            strcpy(aux,A[i]);
            strcpy(A[i],A[i+1]);
            strcpy(A[i+1],aux);
            suntInv=1;}
    if(suntInv==0)break;
    else ultim--;
}
//afisare vector sortat:
for(i=0;i<N;i++)
    printf("\n%s",A[i]);
getch();
}

```

Programul nr. 9

Să se creeze un șir citit de la tastatură, după algoritmul introdus de Louis Mansfiel, în 1936. La acest tip de criptare, fiecare literă este înlocuită cu două vocale. Toate literele alfabetului englez, sunt plasate în celulele unei matrici pătrate de 5 linii și 5 coloane. Pe marginile acestui pătrat se așează cele 5 vocale, astfel:

A E I O U

A a b c d e
 E f g h i/j k
 I l m n o p
 O q r s t u
 U v w x y z

Literele i sau j se codifica la fel.

O literă oarecare din cele 25 din matrice, va fi substituită în textul criptat de cele 2 vocale: cea de pe linie și cea de pe coloana literei din matrice.

Astfel, caracterul x se codează cu perechea de vocale: UI .

Orice alt caracter diferit de litera, se lăsa neschimbat în textul criptat.

Sursa programului:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
void main()
{
    char s[80];
    char sCriptat[160];
    clrscr();
    printf("sir=");
    gets(s);
    //matricea literelor alfabetului (litera j - lipseste. Se
    //va coda separat, ca litera i.
    char C[5][5]={'a','b','c','d','e',
                 'f','g','h','i','k',
                 'l','m','n','o','p',
                 'q','r','s','t','u',
                 'v','w','x','y','z'};
    char V[5]={'A','E','I','O','U'}; //vectorul vocalelor folosit în criptare.
    int c;
    int gasit; //semafor pentru cautarea liniara
    int lin,col;
    int i; //index în sirul s
    int j; //index în sirul sCriptat
    j=0;
    for(i=0; i<strlen(s); i++){
        //este caracterul curent litera?
        c=tolower(s[i]); //Dacă este litera mare, va fi convertit la litera mica.
        //Dacă nu este,
        //va ramane nemodificat.
        if(islower(c)){ //este litera mica?
            //cazul literei j - tratat separat, căci nu este prezentă în matrice:
            if(c=='j'){ //j se codează cu: EO
                sCriptat[j]='E';
                j++;
                sCriptat[j]='O';
                j++;
            }
            else{ //cautăm linia și coloana din matrice de cifrare:
```

```
//cautare liniara:
gasit=0;
for(lin=0;lin<5;lin++){
    for(col=0;col<5;col++){
        if(C[lin][col]==c){
            gasit=1;
            break;}
        if(gasit==1)break;
    }//for
    sCriptat[j]=V[lin];
    j++;
    sCriptat[j]=V[col];
    j++;
} //a fost litera
else{
    //nu este litera:
    sCriptat[j]=c;
    j++;}
} //for i

//terminatorul de sir pentru sirul sCriptat:
sCriptat[j]=0;
printf("%s",sCriptat);
getch();
}
```

4. PROBLEME PROPUSE

1. Se citește un șir de la tastatură. Să se afișeze ultima vocala din șirul citit.

2. Se citește un șir de caractere de la tastatură. Să se rotească tot șirul spre stânga cu o poziție.

Exemplu:

Șirul citit: "abcd"

După rotație: "bcda"

3. Se citește un șir de la tastatură. Să se afișeze care este vocala ce apare de cele mai multe ori în șir.

4. Se citește un șir de caractere de la tastatură. Să se afișeze dacă în șir există sau nu două cifre alăturate.

5. Se citesc două cuvinte de la tastatură. Să se afișeze dacă cel de-al doilea este anagrama primului (o permutare a caracterelor primului șir).

6. Se citește un număr natural de la tastatură. Numărul putând fi foarte mare, citirea lui se va face sub formă de șir. Să se calculeze și afișeze care este cifra maximă din număr.

7. Se citește un șir s1 de la tastatură. Să se construiască șirul s2 ce conține în ordinea apariției, caracterele din șirul s1, luate o singură dată. Exemplu : dacă șirul s1 este : CALCULATOR atunci șirul s2 este : CALUTOR .
