

## Lucrarea de laborator nr. 5 Tema: "Prelucrarea șirurilor de caractere"

### Problema 1

var.	Condiția
1.	Este dat un șir de caractere. De comparat cantitatea vocalelor cu cantitatea consoanelor din șir.
2.	Este dat un șir de caractere. De determinat de câte ori în șir se întâlnește fiecare pereche de caractere identice așezate alături. Exemplu: aa...bb...aa...
3.	Este dat un șir de caractere. De șters din șir toate spațiile de prisos. Astfel, între cuvinte va fi numai câte un spațiu.
4.	Este dat un șir de caractere. De aranjat caracterele din șir în ordine alfabetică.
5.	Este dat un șir de caractere. De aflat de câte ori este întâlnit în șir fiecare din caractere.
6.	Este dat un șir de caractere. De șters din șir toate vocalele.
7.	Este dat un șir de caractere. De șters din șir toate perechile de caractere identice așezate alături. Exemplu: aa, bb.
8.	Este dat un șir de caractere. De determinat valorile și pozițiile ultimelor două vocale din șir.
9.	Este dat un șir de caractere. De schimbat cu locul caracterele unu cu trei, cinci cu șapte, nouă cu unsprezece ș.a.m.d. Dacă ultimul caracter va fi fără pereche, el va rămîne neschimbat.
10.	Este dat un șir de caractere. De indicat simbolurile care se întâlnesc o singură dată în text.
11.	Este dat un șir de caractere. De inversat șirul, folosind principiile clasice de inversare a masivelor.
12.	Este dat un șir de caractere. De șters simbolul " , " din șir și de calculat numărul de înlăturări.
13.	Este dat un șir de caractere. De determinat valorile și pozițiile primelor trei consoane din șir.
14.	Este dat un șir de caractere. Din textul dat de șters caracterele majuscule și de calculat lungimea textului rămas.
15.	Este dat un șir de caractere. De șters din șir caracterele de pe poziții impare.

16.	Este dat un șir de caractere. De șters din șir caracterele minuscule de pe poziții impare.
-----	--

Exemplu de program:

Este dat un șir de caractere. De șters din șir caracterele minuscule de pe poziții impare.

```
#include<string.h>
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main (void) {
char s[256],g[256]; int i,n,z,k;
clrscr(); puts("Culege un sir de caractere\n");
gets(s); n=strlen(s); z=0;
for(i=0;i<n;i++){ k=s[i];
if((k<97)||(k>122)) {g[z]=s[i]; z++;}
if((fmod(i,2)==0)&&(k>=97)&&(k<=122)) {g[z]=s[i]; z++;} }
g[z]='\0'; clrscr();
puts("\nSirul initial s:"); puts(s);
strcpy(s,g);
puts("\nSirul final:"); puts(s);
printf("\nCaracterele minuscule de pe pozitiile impare au fost sterse"); getch(); }
```

## Problema 2

Masive de șiruri. Notă. Șir normal – un șir de caractere compus din cuvinte separate prin spațiu. După ultimul cuvânt urmează punct.

var.	Condiția
1.	Este dat un șir normal. De schimbat cu locul primul și ultimul caracter din fiecare cuvânt al șirului.
2.	Este dat un șir normal. De efectuat căutarea perechilor de cuvinte, în care unul din cuvinte este inversia altuia.
3.	Este dat un șir normal. De aranjat cuvintele din șir în ordine alfabetică.

4.	Este dat un șir normal. De determinat cuvintele cu caracterele aranjate în ordine alfabetică.
5.	Este dat un șir normal. De inversat cuvintele de pe poziții impare.
6.	Este dat un șir normal. De șters litera din mijloc din toate cuvintele de lungime impară.
7.	Este dat un șir normal. De aranjat cuvintele din șir în ordine inversă.
8.	Este dat un șir normal . De determinat cuvântul care conține numărul maximal de consoane.
9.	Este dat un șir normal. De șters din șir cuvintele ce se întâlnesc o singură dată.
10.	Este dat un șir normal. De schimbat cu locul cel mai scurt cuvânt cu cel mai lung cuvânt.
11.	Este dat un șir normal. De șters din șir cuvintele ce se repetă.
12.	Este dat un șir normal. De aranjat cuvintele șirului în ordine descrescătoare după cantitatea de vocale din cuvinte.
13.	Este dat un șir normal. De determinat cuvintele ce se întâlnesc în șir de k ori. Numărul k va fi indicat de utilizator.
14.	Este dat un șir normal. De găsit cuvintele simetrice de lungime pară.
15.	Este dat un șir normal. De șters din șir cuvintele simetrice.
16.	Este dat un șir normal. De inversat cuvintele de lungime pară ce se află pe poziții impare.

Exemplu de program:

Este dat un șir normal. De inversat cuvintele de lungime pară ce se află pe poziții impare.

```
#include<string.h>
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main (void) {
char s[250],G[250]="",x[250][250],SP[2]=" ",PK[2]="."; int n,i,r,k;
clrscr(); puts("Culege un sir de caractere\n");
gets(s); n=strlen(s); r=0; k=0;
for(i=0;i<n;i++) {
```

```

if((s[i]!=' ')&&(s[i]!='.')) {x[r][k]=s[i];k++;}
if((s[i]==' ')/(s[i]=='.')) {x[r][k]='\0';r++;k=0;} }
for(i=0;i<r;i++) {
if( (fmod(i,2)!=0)&&(fmod(strlen(x[i]),2)==0) ) strrev(x[i]); }
for(i=0;i<r;i++) { strcat(G,x[i]);
if(i<(r-1)) strcat(G,SP);}
strcat(G,PK); clrscr();
printf("\nSirul initial:\n"); puts(s); strcpy(s,G);
printf("\nSirul final:\n"); puts(s);
printf("\nCuvintele de lungime para pe pozitii impare au fost inversate\n"); getch(); }

```