Ministerul Educaţiei, Culturii și Cercetării

al Republicii Moldova   
  
Universitatea Tehnică a Moldovei   
  
  
  
  
  
  
  
RAPORT

Lucrarea de laborator nr. 5   
la Programarea Calculatoarelor

A efectuat: st. gr. TI-211 Popa Cătălin   
  
A verificat: Stratulat Ștefan

UTM, Chișinău 2021

**Scopul lucrării**

Scopul lucrării este de a familiariza studentul cu procesarea și utilizarea șirurilor de caractere, utilizînd limbajul C.

**Probleme pentru șiruri de caractere:**

1. Scrieți un program C pentru a afla lungimea unui șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char a[100]="Salut";

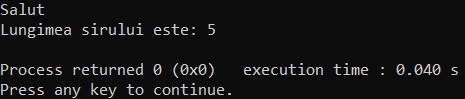
int n;

printf("%s",a);

printf("\nLungimea sirului este: %d\n",strlen(a));

return 0;

}



2. Scrieți un program C pentru a copia un șir în altul.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

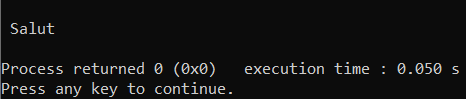
{

char a[100] = "Salut";

char b[100];

printf("\n %s\n",strcpy(b,a));

return 0; }



3. Scrieți un program C pentru a concatena două șiruri.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

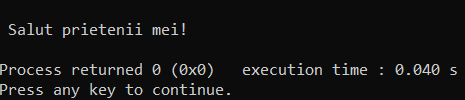
char a[100] = "Salut";

char b[100] = " prietenii mei!";

printf("\n %s\n",strcat(a,b));

return 0;

}



4. Scrieți un program C pentru a compara două șiruri.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str1[100] = "Stratulat";

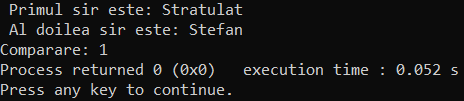
char str2[100] = "Stefan";

printf(" Primul sir este: %s\n Al doilea sir este: %s\n",str1,str2);

printf("Comparare: %d",strcmp(str1,str2));

return 0;

}



5. Scrieți un program C pentru a converti dintr-un șir, literele minuscule în majuscule.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char \*strupr(char \*str)

{

for(int i=0; str[i]; i++)

{

str[i] = toupper(str[i]);

}

}

int main()

{

char str[100] = "omenirea este rea";

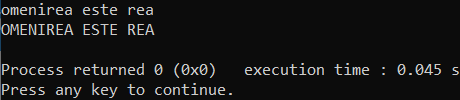
printf("%s\n", str);

strupr(str);

printf("%s\n",str);

return 0;

}



6. Scrieți un program C pentru a converti dintr-un șir, literele majuscule în litere mici.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char \*strupr(char \*str)

{

for(int i=0; str[i]; i++)

{

str[i] = tolower(str[i]);

}

}

int main()

{

char str[100] = "OMENIREA ESTE REA";

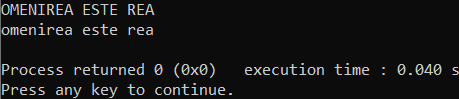
printf("%s\n", str);

strupr(str);

printf("%s\n",str);

return 0;

}



7. Scrieți un program C pentru a comuta literele minuscule în majuscule și majuscule în minuscule pentru fiecare caracter dintr-un șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100];

int i;

printf("\nDati un sir: \n");

gets(str);

for(i=0; str[i]; i++)

{

if(str[i]>='a' && str[i]<='z')

{

str[i]=str[i]-32;

}

else if (str[i]>= 'A' && str[i]<='Z')

{

str[i]=str[i]+32;

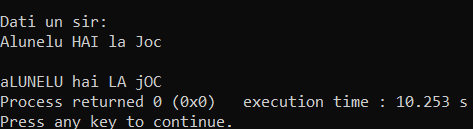
}

}

printf("\n%s",str);

return 0;

}



8. Scrieți un program C pentru a găsi numărul total de litere, cifre sau caractere speciale dintr-un șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100];

int i=0,litere=0,cifre=0,caracter=0;

printf("\nDati un sir: \n");

gets(str);

while(str[i] !='\0')

{

if((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z'))

{

litere++;

}

else if(str[i] >= '0' && str[i] <='9')

{

cifre++;

}

else

{

caracter++;

}

i++;

}

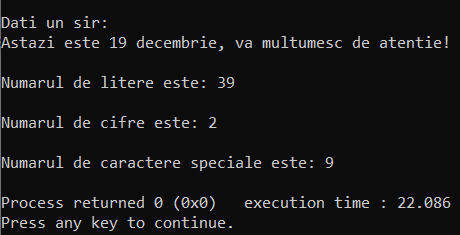
printf("\nNumarul de litere este: %d\n",litere);

printf("\nNumarul de cifre este: %d\n",cifre);

printf("\nNumarul de caractere speciale este: %d\n",caracter);

return 0;

}



9. Scrieți un program C pentru a număra numărul total de vocale dintr-un șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char \*litera(char \*text)

{

int s=0;

for(int i=0; text[i]!='\0'; i++)

{

if(text[i]=='a'||text[i]=='e'||text[i]=='i'||text[i]=='o'||text[i]=='u')

{

s++;

}

}

printf("Numarul vocalelor este: %d\n",s);

}

int main()

{

char text[100];

printf("Dati un text:\n");

gets(text);

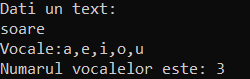
printf("Vocale:a,e,i,o,u\n");

litera(text);

getch();

return 0;

}



10. Scrieți un program C pentru a număra numărul total de consoane dintr-un șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char \*litera(char \*text)

{

int s=0,c=0;

for(int i=0; text[i]!='\0'; i++)

{

if((text[i]>=65 && text[i]<=90)|| (text[i]>=97 && text[i]<=122))

{

if(text[i]=='a'||text[i]=='e'||text[i]=='i'||text[i]=='o'||text[i]=='u')

{

s++;

}

else

{

c++;

}

}

}

printf("Numarul consoanelor este: %d\n",c);

printf("Numarul vocalelor este: %d\n",s);

}

int main()

{

char text[100];

printf("Dati un text:\n");

gets(text);

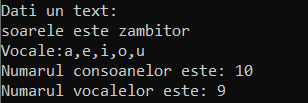
printf("Vocale:a,e,i,o,u\n");

litera(text);

getch();

return 0;

}



11. Scrieți un program C pentru a număra numărul total de cuvinte dintr-un șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char \*litera(char \*str)

{

int s=1,i=0;

while(str[i]!='\0')

{

if(str[i] == ' ' || str[i] == '\n' || str[i] == '\t')

{

s++;

}

i++;

}

printf("Numarul de cuvinte este: %d\n",s-1);

}

int main()

{

char str[100];

printf("Dati un text:\n");

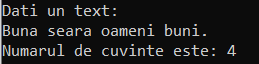
gets(str);

litera(str);

getch();

return 0;

}



12.

Scrieți un program C pentru a găsi inversul unui șir

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char \*litera(char \*str)

{

printf("%s ",strrev(str));

}

int main()

{

char str[100];

printf("Dati un text:\n");

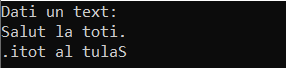
gets(str);

litera(str);

getch();

return 0;

}



13. Scrieți un program C pentru a verifica dacă un șir este palindrom sau nu.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

char \*litera(char \*str)

{

char strinv[100];

strcpy(strinv, str);

strrev(strinv);

if(strcmp(strinv,str) == 0)

printf("Sirul este palindrom\n");

else

printf("Sirul nu este palindrom\n");

}

int main()

{

char str[100];

printf("Dati un text:\n");

gets(str);

litera(str);

getch();

return 0;

}





14. Scrieți un program C pentru a inversa ordinea cuvintelor dintr-un șir dat.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100], reverse[100];

int len, i, x, strstart, strend;

printf("Dati un text: \n");

gets(str);

len = strlen(str);

x = 0;

strstart = len - 1;

strend = len - 1;

while(strstart > 0)

{

if(str[strstart] == ' ')

{

i = strstart + 1;

while(i <= strend)

{

reverse[x] = str[i];

i++;

x++;

}

reverse[x++] = ' ';

strend = strstart - 1;

}

strstart--;

}

for(i=0; i<=strend; i++)

{

reverse[x] = str[i];

x++;

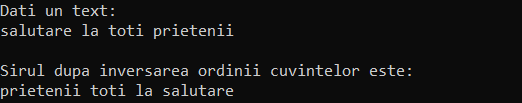
}

reverse[x] = '\0';

printf("\nSirul dupa inversarea ordinii cuvintelor este: \n%s", reverse);

return 0;

}



15. Scrieți un program C pentru a găsi prima apariție a unui caracter într-un șir dat.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100], caracter;

int i, c=0;

printf("Dati un text: \n");

gets(str);

printf("Dati caracterul pe care doriti sa il gasiti: \n");

scanf("%c", &caracter);

for(i = 0; i <= strlen(str); i++)

{

if(str[i] == caracter)

{

c++;

break;

}

}

if(c == 0)

{

printf("Caracterul dat nu a fost gasit in sirul dat '%c' \n", caracter);

}

else

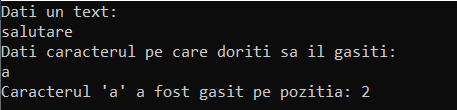
{

printf("Caracterul '%c' a fost gasit pe pozitia: %d \n", caracter, i + 1);

}

return 0;

}



16. Scrieți un program C pentru a găsi ultima apariție a unui caracter într-un șir dat.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100], caracter;

int i, c=0;

printf("Dati un text: \n");

gets(str);

printf("Dati caracterul pe care doriti sa il gasiti: \n");

scanf("%c", &caracter);

for(i = 0; i <= strlen(str); i++)

{

if(str[i] == caracter)

{

c = i;

}

}

if(c == -1)

{

printf("Caracterul dat nu a fost gasit in sirul dat '%c' \n", caracter);

}

else

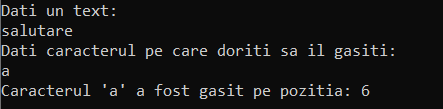
{

printf("Caracterul '%c' a fost gasit pe pozitia: %d \n", caracter, c + 1);

}

return 0;

}



17. Scrieți un program C pentru a căuta toate aparițiile unui caracter dintr-un șir dat.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100], caracter;

int i, c=0;

printf("Dati un text: \n");

gets(str);

printf("Dati caracterul pe care doriti sa il gasiti: \n");

scanf("%c", &caracter);

for(i=0; i<= strlen(str); i++)

{

if(str[i] == caracter)

{

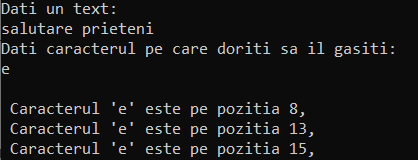
printf("\n Caracterul '%c' este pe pozitia %d, ",caracter, i+1);

}

}

return 0;

}



18. Scrieți un program C pentru a număra aparițiile unui caracter dintr-un șir dat.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100], caracter;

int i, c=0;

printf("Dati un text: \n");

gets(str);

printf("Dati caracterul pe care doriti sa il gasiti: \n");

scanf("%c", &caracter);

for(i=0; i<= strlen(str); i++)

{

if(str[i] == caracter)

{

c++;

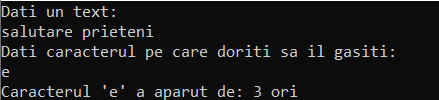
}

}

printf("Caracterul '%c' a aparut de: %d ori",caracter, c);

return 0;

}



19. Scrieți un program C pentru a găsi caracterul cu cea mai mare frecvență dintr-un șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100], caracter;

int i,l,max=-1;

int freq[256] = {0};

printf("Dati un text: \n");

gets(str);

l=strlen(str);

for(i=0; i<l; i++)

{

freq[str[i]]++;

}

for(i=0; i<l; i++)

{

if(max < freq[str[i]])

{

max = freq[str[i]];

caracter=str[i];

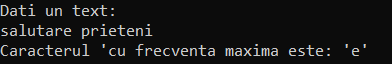
}

}

printf("Caracterul 'cu frecventa maxima este: '%c'",caracter);

return 0;

}



20. Scrieți un program C pentru a găsi caracterul cu frecvența cea mai joasă dintr-un șir.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main()

{

char str[100];

int i=0,l,min,a;

int freq[256] = {0};

printf("Dati un text: \n");

gets(str);

while(str[i] != '\0')

{

a=(int)str[i];

freq[a] += 1;

i++;

}

min=0;

for(i=0; i<256; i++)

{

if(freq[i] != 0)

{

if(freq[min] == 0 || freq[i] < freq[min])

{

min = i;

}

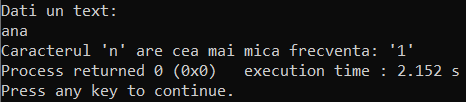
}

}

printf("Caracterul '%c' are cea mai mica frecventa: '%d'",min,freq[min]);

return 0;

}



**Rezultatul lucrării:**

1.ce reprezintă un șir de caractere în limbajul C?

In limbajul C, un sir de caractere reprezinta un tablou de caractere, care are ca ultimul element al sirului caracterul null.

2. cu ce simbol se finisează un șir de caractere în limbajul C?

In limbajul C , un sir se finiseaza cu caracterul null ‘\0’.

3. cum se inițializează un șir de caractere la definirea acestuia ?

Se defineste un tabloul de tip char de dimensiunea necesara ,ca exemplu: char text[25]=”Buna seara”;

4. cum citim de la tastatură un șir de caractere cu spații ?

Un sir de caractere cu spatii se citeste astfel: gets(text);

5. cum aflăm lungimea unui șir de caractere?

Lungimea unui sir de caractere o putem afla cu ajutorull instructiunii: strlen(const chat);

6. cum aflăm numărul de cuvinte într-un șir de caractere?

Initializam o variabila n cu valoarea initiala 0. Citim elementele sirului si in caz ca se ajunge la elementul “ ”-spatiu liber, “\n”-treceerea din rand nou sau “\t” -caracterul special tab, la variabila initiala n cu valoarea zero se aduna o valoare. Astfel in final citim variabila n si aflam numarul de cuvinte.