**Laborator №7 : *Șiruri de caractere***

1. Să se implementeze funcţiile strlen, strcpy, strcmp, strstr, strcat.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <ctype.h>

void \_menu(void)

{

printf("\n------------ Meniu Interactiv ------------\n"

"0. Iesire\n"

"1. Introducerea primului sir\n"

"2. Introducerea sirului secund\n"

"3. Afisarea primului sir\n"

"4. Afisarea sirului secund\n"

"5. Apelarea strlen\n"

"6. Apelarea strcpy(sir1, sir2)\n"

"7. Apelarea strcmp(sir1, sir2)\n"

"8. Apelarea strstr(sir1, subsir)\n"

"9. Apelarea strcat(sir1, sir2)\n\n"

"Optiunea dvs : ");

}

int main(void)

{

char s1[30], s2[30], search[30];

int opt = 1;

do

{

\_menu();

scanf("%d", &opt);

switch (opt)

{

case 0:

exit(0);

break;

case 1:

{

printf("Introuceti primul sir : ");

getchar();

gets(s1);

printf("\n");

}

break;

case 2:

{

printf("Introduceti sirul secund : ");

getchar();

gets(s2);

printf("\n");

}

break;

case 3:

{

printf("Primul sir : ");

puts(s1);

printf("\n");

}

break;

case 4:

{

printf("Sirul secund : ");

puts(s2);

printf("\n");

}

break;

case 5:

{

printf("Lungimea sirului 1 : %d\n", strlen(s1));

printf("Lungimea sirului 2 : %d\n", strlen(s2));

printf("\n");

}

break;

case 6:

{

strcpy(s1, s2);

printf("Sirul 1 (care a copiat continutul din sir 2): %s", s1);

printf("\n");

}

break;

case 7:

{

if (strcmp(s1, s2) > 0)

printf("Sirul 1 < Sirul 2");

else if (strcmp(s1, s2) == 0)

printf("Sirul 1 = Sirul 2");

else if (strcmp(s1, s2) < 0)

printf("Sirul 1 < Sirul 2");

printf("\n");

}

break;

case 8:

{

printf("Introduceti subsirul cautat in sirul 1 : ");

getchar();

gets(search);

if (strstr(s1, search) != NULL)

printf("Sirul cautat se contine in sirul 1");

else

printf("Sirul cautat nu se contine in sirul 1");

printf("\n");

}

break;

case 9:

{

printf("Rezultatul concatenarii sirului 1 si 2 : %s", strcat(s1, s2));

printf("\n");

}

break;

default:

printf("Eroare! Optiune nedefinita !");

break;

}

} while (opt != 0);

return 0;

}

**2)** Să se scrie un program C care să citească de la tastatură un cuvânt şi să verifice dacă respectivul cuvânt este palindrom (cuvânt care poate fi citit de la stânga la dreapta şi de la dreapta la stânga fără să-şi piardă sensul: cojoc, capac, rar).

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <ctype.h>

int \_palindrom(char \*s)

{

int palindrom = 1;

for (int i = 0; i < (strlen(s) / 2); i++)

{

if (s[i] != s[strlen(s) - 1 - i])

{

palindrom = 0;

break;

}

}

return palindrom;

}

int main(void)

{

char s[30];

printf("Introduceti un cuvant, pentru a verifica daca este palindrom : ");

gets(s);

switch (\_palindrom(s))

{

case 0:

printf("Cuvantul introdus nu este palindrom !");

break;

case 1:

printf("Cuvantul introdus este palindrom !");

break;

default:

break;

}

printf("\n");

return 0;

}

1. Să se scrie un program C care realizează următoarele: a) Şterge dintr-un şir de caractere un subşir specificat prin poziţie şi lungime. b) Inserează într-un şir începând cu o poziţie dată un al şir. c) Citeşte doua cuvinte şi înlocuieşte într-un text introdus de la tastatură toate apariţiile primului cuvânt prin cel de-al doilea.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

void \_menu(void)

{

printf("\n1. citirea unui sir de caractere\n"

"2. afisarea sirului de caractere\n"

"3. sterge un subsir specificat prin pozitie si lungime din sir\n"

"4. insereaza un subsir incepand cu o pozitie data a sirului\n"

"5. inlocuiteste toate aparitiile unui cuvant din sir cu altul\n"

"0. iesire\n\n"

"Optiunea dvs : ");

}

int main(void)

{

char s[30], s0[30], s1[30], cuv1[30], cuv2[30], \*index;

int pos, len, i, j, aux, opt = 1, k, m;

do

{

\_menu();

scanf("%d", &opt);

switch (opt)

{

case 0:

exit(0);

break;

case 1:

{

printf("Introdu sirul de caractere : ");

getchar();

gets(s);

printf("\n");

}

break;

case 2:

{

printf("Sirul introdus : ");

printf("%s", s);

printf("\n");

}

break;

case 3:

{

printf("Introdu pozitia : "); scanf("%d", &pos);

printf("Introdu lungimea : "); scanf("%d", &len);

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

{

s1[i] = s[i];

}

for (j = pos + len; j <= strlen(s); j++)

{

s1[i] = s[j];

i++;

}

printf("Sirul nou format : %s\n", s1);

}

break;

case 4:

{

printf("Introdu pozitia : "); scanf("%d", &pos);

printf("Introdu subsirul care urmeaza sa fie inserat in sir : ");

scanf("%s", s1);

char\* ptr = (char\*)malloc(strlen(s) + strlen(s1) + 1);

strncpy(ptr, s, pos);

ptr[pos] = '\0';

strcat(ptr, s1);

strcat(ptr, s + pos);

printf("Sirul nou-format : %s", ptr);

free(ptr);

}

break;

case 5:

{

printf("Introduceti primul cuvant (cautat) : ");

scanf("%s", cuv1);

printf("Introduceti cuvantul secund (inlocuit) : ");

scanf("%s", cuv2);

char\* ptr = strstr(s, cuv1);

if (ptr != NULL)

{

k = 0;

for (i = 0; \*(ptr + i) != ' '; i++)

k++;

m = ptr - s;

char\* ptr\_ = (char\*)malloc(strlen(s) + abs(strlen(s) - strlen(cuv1)) + 1);

strncpy(ptr\_, s, m);

ptr\_[m] = '\0';

strcat(ptr\_, cuv2);

strcat(ptr\_, s + (m + k));

printf("%s", ptr\_);

}

else

printf("Subsirul nu se regaseste in sir!");

printf("\n");

}

break;

default:

printf("Eroare! Optiune nedefinita !");

break;

}

} while (opt != 0);

return 0;

}