# Q1Use pointers to count the following in a string :-

# #chars

# #words

# #uppercase

# #lowercase

# #digits

# #symbols(not spaces)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

char line[60];

printf("Enter the string containing chars with UC and LC ,digits,symbols\n");

gets (line);

char \*p=line;

//NO. OF CHARACTERS

// printf("\nThe number of characters in the string is : %d\n",strlen(p));

int count=0;

for(int i=0;i<60;++i){

if(\*(p+i)=='\0'){

break;

}

else{

count++;

}

}

int character=count;

printf("\nThe number of characters in the string is : %d\n",character);

//COUNTING NO OF WORDS

count=0;

for(int i=0;i<strlen(p);++i){

if(\*(p+i)==' ') count++; /\*counting no of spaces\*/

}

int word = count +1;

printf("\nThe number of words in the string is : %d\n",word); /\*words = no. of spaces + 1\*/

//COUNTING UPPER CASE LETTER(65-->90)

count=0;

for(int i=0;i<strlen(p);++i){

for(int j=65;j<=90;++j){

if(\*(p+i)==j)

count++;

}

}

int uc=count;

printf("\nThe number of upper case letters in the string is : %d\n",uc);

//COUNTING LOWER CASE LETTER(97-->122)

count =0;

for(int i=0;i<strlen(p);++i){

for(int j=97;j<=122;++j){

if(\*(p+i)==j) count++;

}

}

int lc=count;

printf("\nThe number of lower case letters in the string is : %d\n",lc);

//COUNTING DIGITS

count=0;

char a='0';

for(int i=0;i<strlen(p);++i){

for(char a='0';a<='9';++a){

if(\*(p+i)==a)

count++;

}

}

int digit=count;

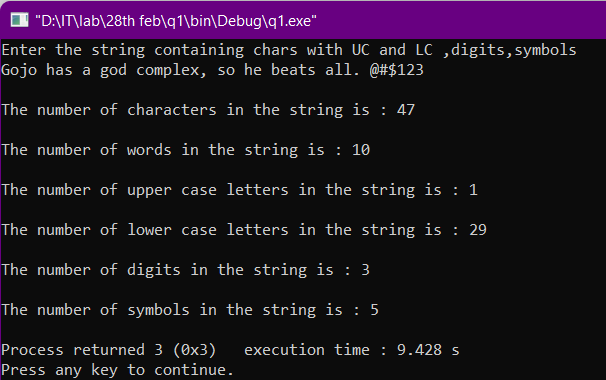
printf("\nThe number of digits in the string is : %d\n",digit);

//COUNTING SYMBOLS

printf("\nThe number of symbols in the string is : %d\n",(strlen(p)-(uc+lc+digit+(word-1))));

return 3;

}



# Q2. Use pointers to delete the first occurrence of a single character in a string.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

int i,j;

char line[60];

printf("Enter a string of characters -->\n");

gets(line);

char \*p=line;

printf("Enter the character to be deleted : \n");

char ch;

ch=getchar();

//SEARCHING THE FIRST OCCURENCE OF CHARACTER CH IN THE STRING

for(i=0;i<strlen(line);++i){

if(line[i]==ch){

for(j=i;j<strlen(line);++j){

\*(p+j)=\*(p+j+1);/\*shifting and deleting array elements\*/

}

break;

}

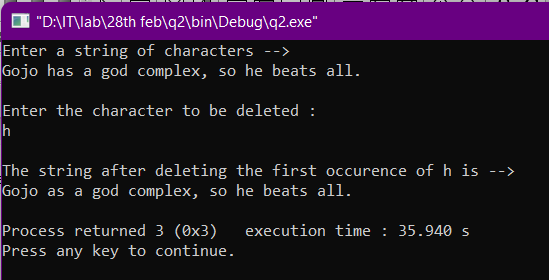
}

printf("\nThe string after deleting the first occurence of %c is -->\n",ch);

puts(line);

return 3;

}



# Q3. Use pointers and delete all spaces in a string.

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <string.h>

int main()

{

int i,j;

printf("Enter a string of characters -->\n");

char line[60];

gets(line);

char \*p=line;

//HERE WE NEED TO REMOVE ALL THE SPACES FROM THE STRING

for(i=0;i<strlen(line);++i){

if(isspace(\*(p+i))){

for(j=i;j<strlen(line);++j){

\*(p+j)=\*(p+j+1);

}

}

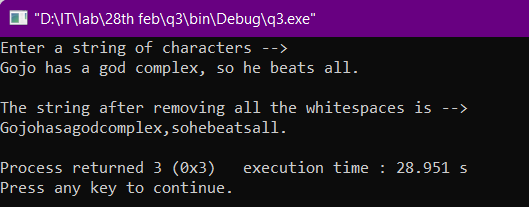
}

printf("\nThe string after removing all the whitespaces is -->\n");

puts(line);

return 3;

}



# Q4(a). Use pointers and find the first occurrence of a substring in a string.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//USE "Gojo has a god complex, so he beats all." FOR INPUT

int main()

{

int i,j;

printf("Enter the string -->\n");

char line[60];

gets(line);

char \*p=line;

printf("\nEnter the substring -->\n");

char sub[20];

gets(sub);

char \*q=sub;

//LOOP FOR FINDING FIRST OCCURENCE OF THE SUBSTRING.

int times=0;

int word=1;

for(i=0;i<strlen(line);++i){

int count=0;

if(\*(p+i)==' ') word++;

for(j=0;j<strlen(sub);++j){

if(\*(p+i+j)==\*(q+j)) count++;

}

if(count==strlen(sub)){

times++;

goto label;

}

}

if(times=0){

printf("\nThe substring was not found.\n");

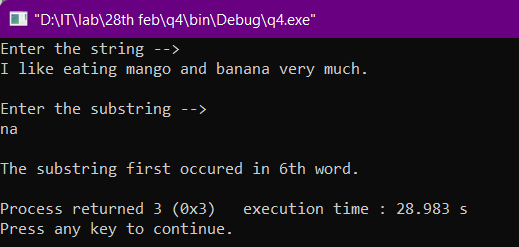
}

label:

printf("\nThe substring first occured in %dth word.\n",word);

return 3;

}



# Q4(b). Use pointers and find the no. of times the substring occurred in a string.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//USE "Gojo has a god complex, so he beats all." FOR INPUT

int main()

{

int i,j;

printf("Enter the string -->\n");

char line[60];

gets(line);

char \*p=line;

printf("\nEnter the substring -->\n");

char sub[20];

gets(sub);

char \*q=sub;

int times=0;

for(i=0;i<strlen(line);++i){

int count=0;

for(j=0;j<strlen(sub);++j){

if(\*(p+i+j)==\*(q+j)) count++;

}

if(count==strlen(sub)){

times++;

}

}

if(times!=0){

printf("\nThe no. of times the substring repeated is : %d\n",times);

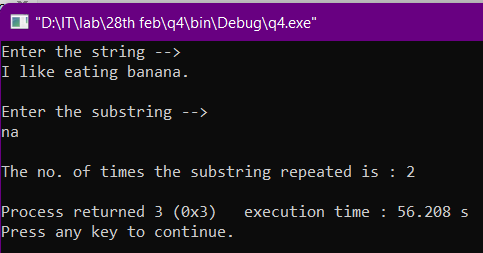
}

else

printf("\nThe substring was not found.\n");

return 3;

}



# Q5. Reverse a string and check if it is a palindrome.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int i;

printf("Enter a string -->\n");

char line[60];

gets(line);

char \*p=line;

printf("\nReverse of the string is -->\n");

for(i=1;i<=strlen(line);++i){ /\*TAKE PRECAUTION FOR NULL CHARACTER\*/

printf("%c",\*(p+strlen(line)-i));

}

printf("\n");

//CHECKING FOR PALINDROME

int count=0;

for(i=0;i<strlen(line);++i){

if(\*(p+i)==\*(p+strlen(line)-i-1)) count++; /\*INTRODUCED -1 IN INDEX ONLY DUE TO NULL CHARACTER\*/

}

if(count==strlen(line))

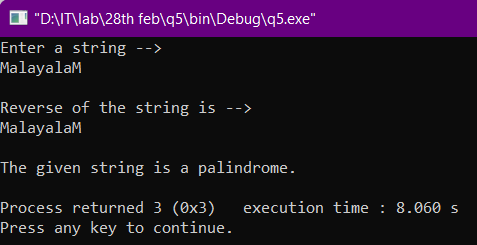
printf("\nThe given string is a palindrome.\n");

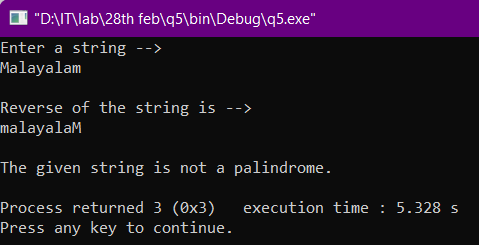
else

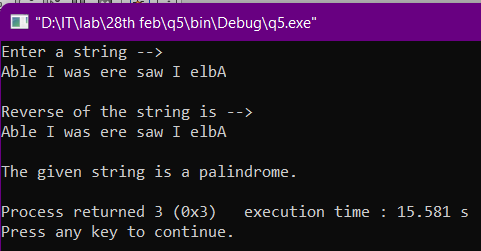
printf("\nThe given string is not a palindrome.\n");

return 3;

}







# Q6. Define the functions on your own:

# \*Strcpy

# \*Strcat

# \*Strcmp

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

void copy(char a1[60],char a2[60])

{

char a3[60];

int i;

for(i=0;a2[i]!='\0';++i){

a3[i]=a2[i];

}

a3[i]='\0';

printf("\nThe string after applying strcpy function is -->\n");

printf("%s",a3);

printf("\n");

}

void cat(char b1[60],char b2[60])

{

int i=0,j=0;

char b3[100];

while(b1[i]!='\0'){

b3[j]=b1[i];

i++;

j++;

}

i=0;

while(b2[i]!='\0'){

b3[j]=b2[i];

i++;

j++;

}

b3[j]='\0';

printf("\nThe string after applying strcpy function is -->\n");

printf("%s",b3);

printf("\n");

}

int cmp(char c1[60],char c2[60])

{

int i=0,j=0,t1=0;

for(i=0;c1[i]!='\0';++i){

if(c1[i]==c2[i]){

t1++;

}

else{

return (c1[i]-c2[i]);

}

}

if(t1==i) return 0;

}

int main()

{

printf("Enter the line in string 1 -->\n");

char s1[60];

gets(s1);

printf("\nEnter the line in string 2 -->\n");

char s2[60];

gets(s2);

copy(s1,s2);

cat(s1,s2);

printf("\nThe string after applying strcmp function is -->\n");

printf("%d",cmp(s1,s2));

printf("\n");

return 3;

}

