1. Palindrome

// Palindrome or not

#include <stdio.h>

#include <string.h>

int main() {

char str[30];

printf("Enter a string:\n");

scanf("%s",str);

char str2[30];

char ch;

int len=strlen(str),flag=0;

strcpy(str2,str);

int j=len-1;

for(int i=0;i<len/2;i++){

ch=str[i];

str[i]=str[j];

str[j]=ch;

j--;

}

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

if(str[i]==str2[i]){

flag=1;

}

}

if(flag==1){

printf("Given string is palindrome");

}

else{

printf("Given string is not palindrome");

} return 0;

}

1. Insert substring to string

// Palindrome or not

#include <stdio.h>

#include <string.h>

int main() {

char str[30];

printf("Enter a main string:\n");

scanf("%s",str);

char substr[30];

char strf[30];

printf("Enter a substring:\n");

scanf("%s",substr);

int pos,i=0;

printf("Enter a position:\n");

scanf("%d",&pos);

int lenm=strlen(str);

int lens=strlen(substr);

for(i=0; i<pos-1;i++){

strf[i]=str[i];

}

int a=i;

int j=0;

for(i=pos-1; i<a+lens;i++){

strf[i]=substr[j];

j++;

}

int k=pos-2;

for(i=pos+lens-1; i<lenm+lens+1;i++){

strf[i]=str[k];

k++;

}

printf("%s",strf);

}

OR

#include <stdio.h>

#include <string.h>

int main() {

char str[30];

printf("Enter a main string:\n");

scanf("%s",str);

char substr[30];

printf("Enter a substring:\n");

scanf("%s",substr);

int pos,i=0;

printf("Enter a position:\n");

scanf("%d",&pos);

int lenm=strlen(str);

int lens=strlen(substr);

int l=lenm+lens;

for(int i=l-1; i>=pos-1;i--){

str[i+lens]=str[i];

}

int j=0;

for(int i=pos-1;i<l;i++ ){

if(j<lens){

str[i]=substr[j];

}

j++;

}

printf("%s",str);

}

1. Tokenized string

#include <stdio.h>

#include <string.h>

int main() {

char str[30]="prerna laxman kote";

char str[100];

int count=0;

char \*token;

token=strtok(str," ");

while(token!=NULL){

count++;

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

token=strtok(NULL," ");

}

printf("No. of words :%d\n",count);

return 0;

}

1. Substring is present or not

#include <stdio.h>

#include <string.h>

int ischeck(char []);

int acheck(char []);

int ischeck(char str1[]){

int len=strlen(str1);

int i;

int flag=0;

for( i=0;i<len;i++){

if(i==0){

if(str1[i]=='i' && str1[i+1]=='s' && (str1[i+2]==' ' || str1[i+2]=='\0')){

flag=1;

break;

}

}

else{

if(str1[i-1]==' ' && str1[i]=='i' && str1[i+1]=='s' && (str1[i+2]==' ' || str1[i+2]=='\0')){

flag=1;

break;

}

}

}

if(flag==0){

return -1;

}

else{

return i;

}

}

int acheck(char str2[]){

int len=strlen(str2);

int flag=0;

int i;

for( i=0;i<len;i++){

if(i==0){

if(str2[i]=='a' && (str2[i+1]==' ' ||str2[i+1]=='\0')){

flag=1;

break;

}

}

else{

if(str2[i-1]==' ' && str2[i]=='a' && (str2[i+1]==' ' ||str2[i+1]=='\0')){

flag=1;

break;

}

}

}

if(flag==0){

return -1;

}

else{

return i;

}

}

int main() {

char str[30]="there is a";

int res1=ischeck(str);

printf("%d\n",res1);

int res2=acheck(str);

printf("%d\n",res2);

return 0;

}

1. Check anagram

// Online C compiler to run C program online

#include <stdio.h>

#include <string.h>

void checkanagram(char [],char[]);

void checkanagram(char str1[],char str2[]){

int l1=strlen(str1);

int l2=strlen(str2);

char temp;

int count=0;

if(l1==l2){

for(int i=0;i<l1; i++){

for(int j=i+1;j<l1;j++){

if(str1[i]>=str1[j]){

temp=str1[i];

str1[i]=str1[j];

str1[j]=temp;

}

if(str2[i]>=str2[j]){

temp=str2[i];

str2[i]=str2[j];

str2[j]=temp;

}

}

if(str1[i]==str2[i]){

count++;

}

}

if(count==l1){

printf("Strings are anagram");

}

else{

printf("Strings are not anagram");

}

}

else{

printf("Strings are not anagram");

}

}

int main() {

char str1[30], str2[30];

printf("Enter first string:\n");

fgets(str1, sizeof(str1), stdin);

printf("Enter second string:\n");

fgets(str2, sizeof(str2), stdin);

checkanagram(str1,str2);

return 0;

}

1. Uppercases in left and right partition

#include <stdio.h>

#include <string.h>

int main() {

char str[30];

printf("Enter first string:\n");

fgets(str, sizeof(str), stdin);

int l1=strlen(str);

int i,j,k, par=0;

for(i=0;i<l1-2;i++){

int countl=0,countr=0;

for(k=0; k<=i;k++){

if(str[k]>='A' && str[k]<='Z'){

countl++;

}

}

for(int j=i+1;j<l1;j++){

if(str[j]>='A' && str[j]<='Z'){

countr++;

}

}

if(countl>countr){

par++;

} }

printf("%d",par);

}

1. */\**
2. *\* C Program to Sort Word in String*
3. *\*/*
4. #include <stdio.h>
5. #include <string.h>
7. void main()
8. {
9. int count = 0, c = 0, i, j = 0, k, l, space = 0;
10. char str[100], p[50][100], str1[20], ptr1[50][100], cmp[50];
12. printf("Enter the string**\n**");
13. scanf(" %[^**\n**]s", str);
14. for (i = 0;i < strlen(str);i++)
15. {
16. if ((str[i] == ' ')||(str[i] == ', ')||(str[i] == '.'))
17. {
18. space++;
19. }
20. }
21. for (i = 0, j = 0, k = 0;j < strlen(str);j++)
22. {
23. if ((str[j] == ' ')||(str[j] == 44)||(str[j] == 46))
24. {
25. p[i][k] = '**\0**';
26. i++;
27. k = 0;
28. }
29. else
30. p[i][k++] = str[j];
31. }
32. for (i = 0;i < space;i++) *//loop for sorting*
33. {
34. for (j = i + 1;j <= space;j++)
35. {
36. if ((strcmp(p[i], p[j]) > 0))
37. {
38. strcpy(cmp, p[i]);
39. strcpy(p[i], p[j]);
40. strcpy(p[j], cmp);
41. }
42. }
43. }
44. printf("After sorting string is **\n**");
45. for (i = 0;i <= space;i++)
46. {
47. printf("%s ", p[i]);
48. }
49. }

Sorted words in a sentence:

// Online C compiler to run C program online

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int compare(const void\* a, const void\* b){

return strcmp(\*(const char \*\*)a,\*(const char \*\*)b);

}

int main() {

char str[100];

char \*word[100];

int count=0;

printf("enter a sentence:");

fgets(str, sizeof(str),stdin);

//remove new line character

str[strcspn(str,"\n")]='\0';

char \*token=strtok(str,"|");

while(token!=NULL && count<100){

word[count++]=strdup(token);

token=strtok(NULL,"|");

}

qsort(word, count,sizeof(char \*), compare);

printf("Sorted Sequence:\n");

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

printf("%s",word[i]);

if(i<count-1){

printf("|");

}

}

}

First non repeating character in a string

#include <stdio.h>

#include <string.h>

#define MAX 100

int main()

{

char str[MAX];

printf("Enter the string : ");

gets(str);

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

{

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

}

int count[26]={0};

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

{

count[str[i]-'a']++;

}

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

{

if(count[str[i]-'a'] == 1)

{

printf("%c\n",str[i]);

return 0;

}

}

printf("All the characters are repeated\n");

return 0;

}

* Capitalized string

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void cap(char \*str){

int issmall=1;

while(\*str){

if((\*str>='a' && \*str<='z') || (\*str>='A' && \*str<='Z')){

if(issmall){

if(\*str>='a' && \*str<='z'){

\*str-=('a'-'A');

}

issmall=0;

}

}

else{

issmall=1;

}

str++;

}

}

int main() {

char str[100];

printf("Enter a string:\n");

fgets(str,sizeof(str),stdin);

str[strlen(str)-1]='\0';

cap(str);

printf("Modified string is:%s\n",str);

}

* Delete duplicate character

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void deletedup(char \*str){

int i,j;

for(i=0,j=0;str[i]!='\0';i++){

if(str[i]==str[i+1]){

do{

i++;

}

while(str[i]==str[i+1]);

}

else{

str[j]=str[i];

j++;

}

}

str[j]='\0';

}

int main() {

char str[100],sub[100];

gets(str);

deletedup(str);

printf("%s",str);

}

* Delete consecutive characters in a string

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void consecutive(char str[]){

int i=0,j,k,count=0;

int len=strlen(str);

while(str[i]!='\0' && str[i+1]!='\0'){

if(str[i+1]-str[i]==1){

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

str[j]=str[j+2];

}

count++;

i=0;

}

else{

i++;

}

}

//str[j]='\0';

printf("Modified string is:%s\n",str);

printf("%d",count);

}

int main() {

char str[100];

printf("Enter a string:\n");

fgets(str,sizeof(str),stdin);

str[strcspn(str,"\n")]='\0';

consecutive(str);

}

* Delete substring

#include<stdio.h>

#include<string.h>

int main()

{

int i, j = 0, k = 0,n = 0,l;

int flag = 0;

char str[100], neww[100], word[100];

printf("Enter Any String to Remove a Word from String: ");

gets(str);

printf("\n\n Enter Any Word You Want to be Removed: ");

gets(word);

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

{

k = i;

while(str[i] == word[j])

{

i++;

j++;

if(j == strlen(word))

{

if(str[k-1]==' ' && str[i]==' '){

flag = 1;

break;

}

}

}

j = 0;

if(flag == 0)

i = k;

else

flag = 0;

neww[n++] = str[i];

// printf("%d",str[i]);

}

neww[n] = '\0';

printf("\n\n After Removing Word From String: %s",neww);

}

* Strspn

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int strspn1(char \*str,char \*strf){

int i=0,count=0;

while(\*str!='\0'){

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

if(\*str==strf[i]){

count++;

break;

}

i++;

}

if(strf[i]=='\0'){

break;

}

str++;

}

return count;

}

int main() {

char str[100],strf[100];

gets(str);

gets(strf);

int res=strspn1(str,strf);

printf("%d",res);

}

* Print anagram from sentence

#include<stdio.h>

#include<string.h>

int isanagram(const char \*word,const char \*nextword){

int count[256]={0};

int i=0;

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

count[(int)word[i]]--;

i++;

}

i=0;

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

count[(int)nextword[i]]--;

i++;

}

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

if(count[i]!=0){

return 0;

}

}

return 1;

}

void display(char str[]){

char str1[100];

strcpy(str1,str);

char \*word=strtok(str1," ");

while(word!=NULL){

char \*nextword=strtok(NULL," ");

while(nextword!=NULL){

if(isanagram(word,nextword)){

printf("%s\t%s",word,nextword);

}

nextword=strtok(NULL," ");

}

word=strtok(NULL," ");

}

}

int main()

{

char str[100];

printf("Enter string:\n");

fgets(str,sizeof(str),stdin);

str[strlen(str)-1]='\0';

display(str);

}

* Squeeze

#include<stdio.h>

#include<string.h>

int main()

{

char str[100],c;

printf("Enter sentence:\n");

fgets(str,sizeof(str),stdin);

printf("Enter target:\n");

scanf("%c",&c);

str[strlen(str)-1]='\0';

for(int i=0,j=0;str[i]!='\0';i++){

if(str[i]!=c){

str[j++]=str[i];

}

str[j]='\0';

}

printf("%s",str);

}

* Compare Date

// Online C compiler to run C program online

#include <stdio.h>

struct date{

int day;

int month;

int year;

};

int comparedate(struct date date1, struct date date2){

}

int main() {

int n;

printf("Enter number of pairs:\n");

scanf("%d",n);

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

printf("Enter first date:\n");

scanf("%d %d %d",&date1.day,&date1.month,&date1.year);

printf("Enter second date:\n");

scanf("%d %d %d",&date2.day,&date2.month,&date2.year);

int result=comparedate(date1,date2);

if(result==-1){

printf("Date1 is earlier than date2");

}

else if(result==1){

printf("Date1 is late than date2");

}

else{

printf("Dates are equal");

}}

return 0;

}