/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Preliminary :: Write C functions ( userdefined) to implement basic string manipulations such as

string length, string compare, string concatination, finsing sub string etc

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<stdio.h>

int length(char str[])

{

int i=0;

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

{

i++;

}

return i;

}

void main()

{

char str[20];int n;

printf("enter the string\n");

scanf("%s",str);

n=length(str);

printf("length of string is %d",n);

}

#include<stdio.h>

#include<string.h>

int compare(char str1[],char str2[])

{

int i=0,flag=0;

while(str1[i]!='\0' && str2[i]!='\0')

{

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

break;

i++;

}

if(str1[i]=='\0' && str2[i]=='\0')

flag=1;

else

flag=0;

return flag;

}

void main()

{

char str1[20],str2[20];int n;

printf("enter the first string\n");

scanf("%s",str1);

printf("enter the second string\n");

scanf("%s",str2);

n=compare(str1,str2);

if(n==1)

printf("strings are equal\n");

else

printf("strings are not equal\n");

}

#include<stdio.h>

#include<stdlib.h>

void con\_cat(char a[],char b[])

{

int i=0,j=0;

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

{

i++;

}

while(b[j]!='\0')

{

a[i+j]=b[j];

j++;

a[i+j]='\0';

}

puts(a);

}

void main()

{

char a[30];

char b[20];

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

scanf("%s",a);

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

scanf("%s",b);

con\_cat(a,b);

}

#include<stdio.h>

#include<string.h>

int search(char[], char[]);

int main() {

int loc;

char str[];

char sub[];

printf("enter a string\n");

scanf("%s",str);

printf("enter a substring\n");

scanf("%s",sub);

loc = search(source, target);

if (loc == -1)

printf("\nNot found");

else

printf("\nFound at location %d", loc + 1);

return (0);

}

int search(char src[], char str[]) {

int i, j, firstOcc;

i = 0, j = 0;

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

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

i++;

if (src[i] == '\0')

return (-1);

firstOcc = i;

while (src[i] == str[j] && src[i] != '\0' && str[j] != '\0') {

i++;

j++;

}

if (str[j] == '\0')

return (firstOcc);

if (src[i] == '\0')

return (-1);

i = firstOcc + 1;

j = 0;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Write a program to reverse individual words in a given line of string

Input : india is great

output : aidni si taerg

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include <string.h>

void reverse\_string(char\*);

void reverse\_words(char\*);

void main() {

char a[100];

printf("enter a string\n");

gets(a);

reverse\_words(a);

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

}

void reverse\_words(char \*s) {

char b[100], \*t, \*z;

int c = 0;

t = s;

while (\*t)

{

while (\*t != ' ' && \*t != '\0')

{

b[c] = \*t;

t++;

c++;

}

b[c] = '\0';

c = 0;

reverse\_string(b);

z = b;

while (\*z)

{

\*s = \*z;

z++;

s++;

}

while (\*s == ' ')

{ // skipping space

s++;

}

t = s; // pointing to next word

}

}

//Function to reverse a word.

void reverse\_string(char \*t)

{

int l, s,e;

char temp;

l = strlen(t);

e = l - 1;

s = 0;

while (s < e)

{

temp = t[s];

t[s] = t[e];

t[e] = temp;

s++;

e--;

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Write a program to check if two given String is the anagram of

each other( string is saved in a structure)

struct str

{

char cont[100];

};

An anagram is a word or phrase formed by rearranging the letters of a different word or phrase,

typically using all the original letters exactly once.

Example

Race =care

Eleven plus two = Twelve plus one

Note inbuilt functions strcmp,strlen & strcpy can be used \*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*