**String(character Array)**

String is nothing but 1-D character array (char array)

**Null character** – ‘\0’ , nothing but a character having ASCII value equal to zero and used as a terminator.

#include <iostream>

using *namespace* std ;

*int* main(){

   // making array of char

*char* name[20];

  cout<<"Enter your name : ";

   cin>>name;

   cout<<"Entered name is : "<<name<<"\n";

}

Test case 1 :

Input

Enter your name : gurjot

output

Entered name is : Gurjot

Test case 2 :

Input

Enter your name : Gurjot Singh

output

Entered name is : Gurjot

Note : There is problem when we took input with the cin in case of char array it stop is execution when there is space , null character , pressed enter and pressed tab.

e.g.

In above test case 2 : we have given input Gurjot Singh but it has only store or pick Gurjot because there is a space b/w Gurjot and Singh .

#include <iostream>

using *namespace* std ;

*int* main(){

   // making array of char

*char* name[20];

   cout<<"Enter your name : ";

   cin>>name;

   // placing null character

   name[2]='\0';

   cout<<"Entered name is : "<<name<<"\n";

}

Character array for above code is : Gu\0rjot\0

Enter your name : Gurjot

Entered name is : Gu

Due to null character placed at 2 index , only Gu is printed

Length of character Array or string

#include <iostream>

using *namespace* std ;

*int* getLength(*char* *str*[])

{

*int* count =  0 ;

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

    {

        count++;

    }

    return count;

}

*int* main(){

*char* str[100];

 cin>>str;

 cout<<"Length of the String is : "<< getLength(str)<<"\n";

}

Gurjot

length of the string is : 6

Reverse a string

*void* reverse(*char* *str*[], *int* *n*)

{

*int* s = 0 ;

*int* e = n-1;

    while(s<e)

    {

        // swap str[s] with str[e]

*char* temp = str[s];

        str[s]=str[e];

        str[e]=temp;

        s++;

        e--;

    }

}

Check Palindrome for a string

#include <iostream>

#include <string.h>

using *namespace* std;

*bool* checkPalindrome(*char* *str*[], *int* *n* )

{

*int* s = 0 ;

*int* e = n-1 ;

    while(s<e)

    {

        if (str[s++]!=str[e--])

        return false ;

    }

    return true;

}

*int* main(){

*char* str[200];

   cin>>str;

   if (checkPalindrome(str, strlen(str)))

   {

    cout<<"Yes string is palindrome"<<endl;

   }

   else

   {

    cout<<"string is not palindrome"<<endl;

   }

}

Uppercase to lowercase or lowercase to uppercase

#include <iostream>

#include <string.h>

using *namespace* std;

*char* toLowercase(*char* *ch*)

{

    // if the given character is already in lowercase

    if (ch>='a' && ch<='z')

    {

        return ch ;

    }

    else{

        return ch-'A'+'a';

    }

}

*char* toUppercase(*char* *ch*)

{

    // if the given character is already in uppercase

    if (ch>='A' && ch<='Z')

    {

        return ch ;

    }

    else{

        return ch-'a'+'A';

    }

}

*int* main(){

*char* ch;

 cin>>ch;

 cout<<"lowercase : "<<toLowercase(ch)<<endl;

 cout<<"Uppercase : "<<toUppercase(ch)<<endl;

}

String palindrome

*bool* palindrom(*int* *n*, string *str*)

{

*int* s = 0 , e=n-1 ;

    while(s<=e)

    {

        if (str[s]!=str[e])

        return false;

        s++,e--;

    }

    return true;

}

Minimum character added to make string Palindrome

#include <iostream>

using *namespace* std ;

*int* minimumCharacterToMakePalindrome(string *str*)

{

*int* start = 0 ;

*int* end = str.length()-1;

*int* sum=0;

    while(start<end)

    {

        if (str[start]!=str[end])

        {

             sum++;

             start++;

        }

        else

        {

            start++;

            end--;

        }

    }

    return sum ;

}

*int* main(){

*int* n ;

   cin>> n ;

   string s ;

   cin>>s  ;

   cout<<mini(s);

}