Harsh Dixit

```
/* CHAR ARRAYS

    CREATION: char array_name[size];
              char ch[10]'

    ACCESS:  ch[index];

    INPUT:   cin >> ch;

    in char arrays we need not to use the loops for taking input the characters
    one by one.

    NULL CHARACTER: it represents the termination of the string.
                    Null character is represented as '\0'
                    its ascii value is 0

    OUTPUT:  cout << ch;

    As like input we can also see ouput just by cout statement.

NOTE ---> char arrays is also passed by refrence.

--------------------------------------------------------------------------------*/

//#include<iostream>
//using namespace std;
//int main()
//{
//     char ch[10];
//     cin >> ch;
//     cout << "The One and Only: " << ch << endl;
//     cout << "ASCII value is: " << (int)ch[6];
//     return 0;
//}

/* DELIMITER -> It is a special character which indicates the beginning or end
                of a statement or string.

    Delimiter of cin in char arrays and strings

    new line character --> '\n'

    tab  -> '\t'
    space  --> '_'


cin.getline()--> when we want to use space and tab while entering the data, we use getline function
                 to take input.

                              cin.getline(array_name,size);
                              cin.getline(ch,100);


       Delimiter of cin.getline()  --> '\n'

--------------------------------------------------------------------------------------------------------------*/
// Length of a string

#include<iostream>
using namespace std;
int findlength(char ch[], int size)
{
       int len = 0;
       while(ch[len] != '\0')
       {
              len++;
       }
       return len;
}
int main()
{
       char ch[100];
       cin.getline(ch,100);

       int length = findlength(ch,100);

       cout << "Length of string is: " << length;

       return 0;

       //strlen() is the inbuild function used for finding string length.
}

//------------------------------------------------------------------------------------

// REVERSE OF A STRING
```

```
#include<iostream>
#include<cstring>
using namespace std;

void reversestring(char ch[] )
{
	int i = 0;
	int j = strlen(ch) - 1;

	while(i <= j)
	{
		swap(ch[i] , ch[j]);
		i++;
		j--;
	}
}

int main()
{
	char ch[100];
	cin.getline(ch,100);

	reversestring(ch);
    cout << "Printing Reverse: " << ch;
	return 0;

	//TIME COMPLEXITY: O(n)

	// strrev() is the inbuilt function used for reverse a string.
}

 //---------------------------------------------------------------------------------------

 // UPPER CASE CONVERSION ->

#include<iostream>
#include<string>
using namespace std;

void uppercase(char ch[] )
{
	int index = 0;
	while(ch[index] != '\0')
```

```
        {
                if(ch[index] >= 'a' && ch[index] <= 'z')
                {
                        ch[index] = ch[index] -'a' + 'A';
                }
                index++;
        }
}

int main()
{
        char ch[100];
        cin.getline(ch,100);

        uppercase(ch);
        cout << "Upper case: " << ch;

        // TIME COMPLEXITY: O(n)
        return 0;
}

//---------------------------------------------------------------------------------------

// REPLACE CHARACTER

#include<iostream>
#include<string>
using namespace std;

void replacechar(char ch[])
{
    int i = 0;

    while(ch[i] != '\0')
    {
       if(ch[i] == '@')
       {
               ch[i] = ' ';
               }
               i++;
       }
}

int main()
```

```
{
       char ch[100];
       cin.getline(ch,100);

    replacechar(ch);

    cout << "After replacment: " << ch;

    //TIME COMPLEXITY -> O(n)
       return 0;
}

//---------------------------------------------------------------------------------------

// CHECK PALINDROME  EX- NOON , CIVIC , RADAR ETC.

#include<iostream>
#include<cstring>
using namespace std;

bool palindrome(char ch[])
{
    int i = 0;
    int j = strlen(ch) - 1;

    while(i <= j)
    {
       if(ch[i] == ch[j])
       {
              i++;
                  j--;
              }
              else
              {
                     return false;
              }
       }
       return true;
}

int main()
{
       char ch[100];
       cin.getline(ch,100);
    bool ispalindrome = palindrome(ch);

    if(ispalindrome == 1)
    {
       cout << "Palindrome..";
       }
       else
       {
              cout << "Not a Palindrome..";
       }

    return 0;
    //TIME COMPLEXITY -> O(n)
       return 0;
}

//-------------------------------------------------------------------------------------
```