**10. Character Arrays ( char)**

“ Arrays of characters are known as Character Arrays.”

**Declaration**

To declare a character array of n size, we do

char arr[n+1];

**Note**: We declare an array of n+1 as 0 to n-1 indices store the actual string and nth character stores **‘\0’** (null character).

# Taking input and Printing output

**Q.(IMP):-**Check if a given character array is a palindrome or not.

Given a string s, on reversing the string we get the same string we call that string is a palindrome.

**Algorithm:**

* Let the length of the character array be n.
* Keep a boolean variable ans to store the result and initialize it with true.
* Iterate over the string and check if **ith** character is equal to **(n-i-1)th**, there can be 2 cases
  + If equal, then do nothing
  + If unequal, then put ans = false
* When the loop ends, if ans is true, then the string is palindrome else it is not a palindrome.

#include<bits/stdc++.h>

using namespace std;

int main(){

    int n;

    cin>>n;

    char arr[n+1];

    cin>>arr;

    bool check = 1;

    for (int i=0; i<n; i++){

        if(arr[i] != arr[n-1-i]){

            check = 0;

            break;

        }

    }

    if(check == true)

      cout<<"word is a Palindrome"<<endl;

    else

        cout<<"word is not a Palindrome"<<endl;

    return 0;

}

**Question**: Largest word in a sentence.

To input a complete sentence, we use the getline() function.

cin.getline(arr, n);

where arr is the character array and n is the total length of sentence

**Approach**

1. Iterate over the sentence and keep variables **currLen** and **maxLen** which store the current length of the present word being iterated and the overall maximum length word’s length.

2. Whenever we encounter a space during iteration, we will maximize our **maxLen** variable.

**maxLen** = max(**maxLen**, **currLen**)

#include<bits/stdc++.h>

using namespace std;

/\*  Large Worde in Sentence of char array   \*/

int main()

{   int n;

    cin>>n;

    cin.ignore();

    char arr[n+1];

    cin.getline(arr, n); //in-build function used

    cin.ignore();

    int i=0;

    int currLen = 0, maxLen = 0;

    int st=0, maxst=0;

    while(1){

        if(arr[i] == ' ' || arr[i] == '\0'){

            if(currLen > maxLen){

                maxLen = currLen;

                maxst = st;

            }

            currLen = 0;

            st = i+1;

        }

        else

        currLen++;

        if(arr[i] == '\0')

            break;

        i++;

    }

    cout<<maxLen<<endl;

    for(int i=0; i<maxLen; i++){

        cout<<arr[i+maxst];

    }

    return 0;

}