CHAR ARRAYS AND STRINGS - LEVEL 1

23 September 2023 08:51

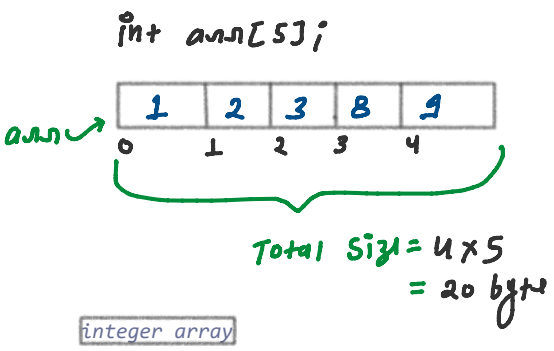
CLASS-14 - 22/09/2023

https://www.linkedin.com/in/manojoffcialmj/

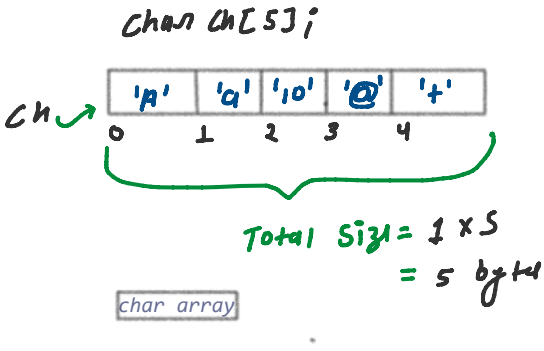
# CHAR ARRAYS & STRINGS LEVEL-1

## ✅ CHAR ARRAYS

### 1. What is char array




integer array




char array

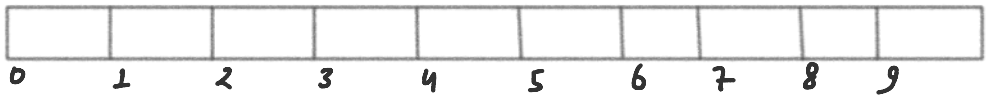
### 2. ASCII CHARACTER CODE 256

*In total, there are 256 ASCII characters, and can be broadly divided into three categories:*

1. *ASCII control characters (0-31 and 127)*
2. *ASCII printable characters (32-126)*
3. *ASCII characters (128-255)*

### 3. Char array creation

```
char ch[10]; // char array creation
```



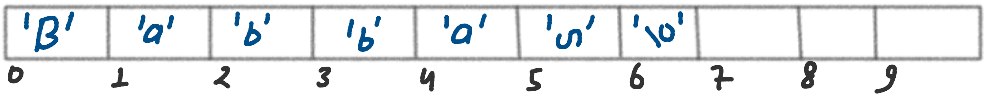

### 4. Taking input in char array

```cpp
char ch[10]; // char array creation

cin>>ch; // taking input Babbar
```

way:01

```cpp
char ch[10]; // char array creation

for(int i=0; i<10; i++){
  cin>>ch[i]; // taking input via index using loop
}
```

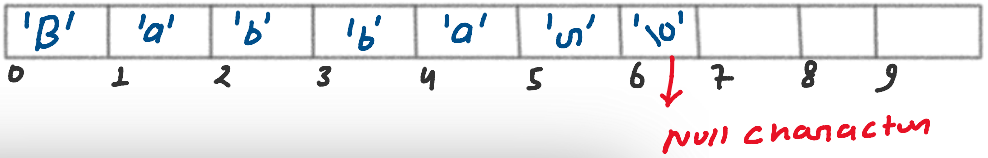
way:2



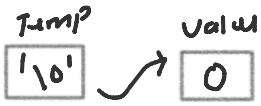

## 5. Print and access char array

```cpp
char ch[10]; // char array creation

cin>>ch; // taking input Babbar

cout<<"Printing the value of ch: "<<ch; // print ch

// access character array's element
cout<<ch[0]; // B
cout<<ch[1]; // a
cout<<ch[2]; // a
cout<<ch[3]; // b
cout<<ch[4]; // a
cout<<ch[5]; // r
cout<<ch[6]; // white space
```

## 6. Null char ASCII CODE




```cpp
char ch[10]; // char array creation

cin>>ch; // taking input Babbar

cout<<"Printing the value of ch: "<<ch; // print ch

// access character array's element
cout<<ch[0]; // B
cout<<ch[1]; // a
cout<<ch[2]; // a
cout<<ch[3]; // b
cout<<ch[4]; // a
cout<<ch[5]; // r
cout<<ch[6]; // null character
```




```
Printing char ASCII CODE: 0
```

```
cout<<ch[3]; // b
cout<<ch[4]; // a
cout<<ch[5]; // r
cout<<ch[6]; // null character

// null char ASCII CODE
char temp = ch[6];
int value = (int)(temp);
cout<<"Printing char ASCII CODE: "<<value;
```

```
Printing char ASCII CODE: 0
```

## 7. Delimiter concept

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

int main() {
    char ch[10]; // char array creation

    cin>>ch; // taking input Love Babbar

    cout<<"Printing the value of ch: "<<ch; // printing ch

    return 0;
}

OUTPUT:
Input: Love Babbar
Printing the value of ch: Love
```

Input

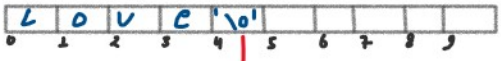
Output

Love \t Tab

Love \n New line

Love \- space

Love ↵ Hit Enter




Ruk jao....

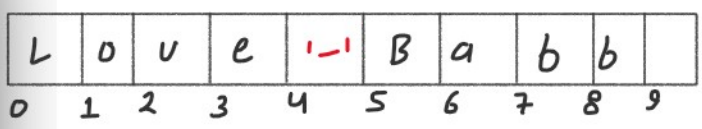
## 8. cin.getline(p1,p2) method

cin.getline( p1 , p2 );

p2 → Char array's capacity

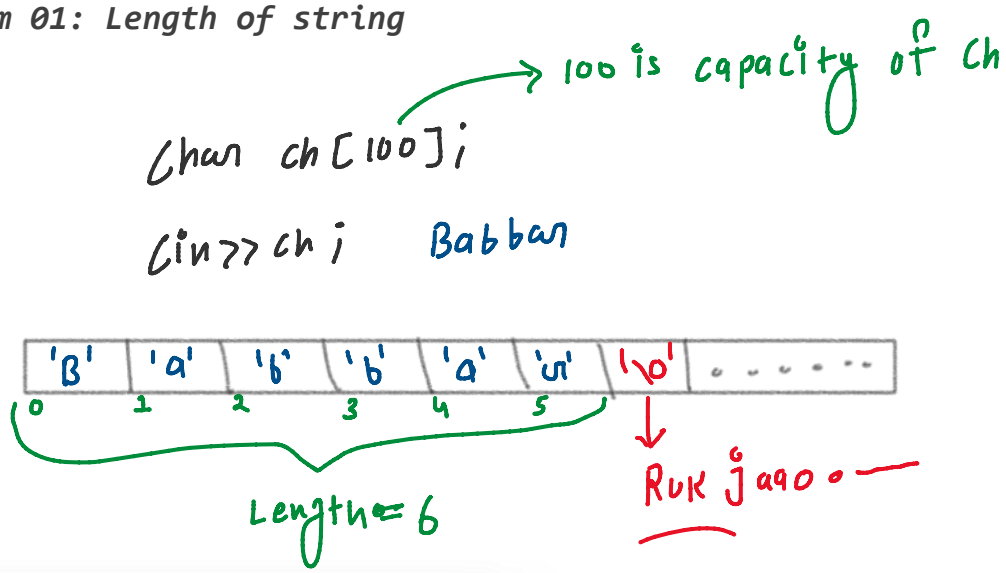
p1 → Input data kis variable memory me store karna chate hai

```
// Online C++ compiler to run C++ program online
#include <iostream>
using namespace std;

int main() {
    char ch[10]; // char array creation

    cin.getline(ch,10); // taking input Love Babbar

    cout<<"Printing the value of ch: "<<ch; // printing ch

    return 0;
}


Input: Love Babbar
Output: Printing the value of ch: Love Babb
```



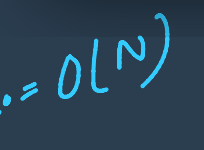

Why? Ye output kyun print ho raha hai?

## 9. Program 01: Length of string




```cpp
// Program 01: Length of string
#include<iostream>
#include <cstring>
using namespace std;

// find length of string
int findLength(char ch[],int size){
    int length = 0;

    for(int i = 0; i < size; i++){
        if(ch[i] == '\0'){
            break;
        }
        else{
            length++;
        }
    }
    return length;
}

int main(){
    char ch[100];

    cin>>ch;

    int length = findLength(ch, 100);

    cout<<"Length of string via our method: "<<length<<endl;
    cout<<"Length of string via builtin method: "<<strlen(ch)<<endl;

    return 0;
}

/*
INPUT: MyNameIsBabbar
OUTPUT:
Length of string via our method: 14
Length of string via builtin method: 14
*/
```

```cpp
// find length of string with the help of while loop
int findLengthByWhileLopp(char ch[],int size){
    int index = 0;

    while(ch[index]!='\0'){
        index++;
    }
    return index;
}
```




## 10. Program 02: Reverse string

TC ⇒ O(N)
SC ⇒ O(1)

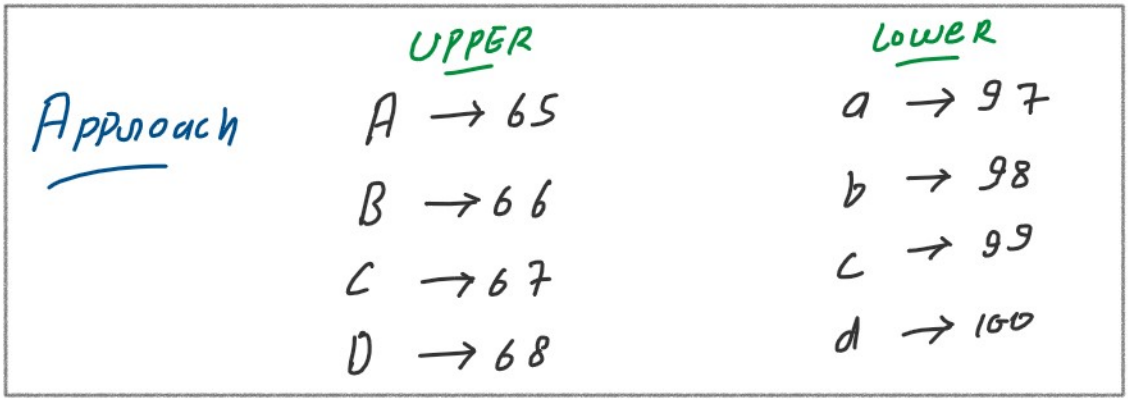
```cpp
// Program 02: Reverse string
#include<iostream>
#include <cstring>
using namespace std;

// Step 01: Find length of string
int findLength(char ch[],int size){
    int index = 0;

    while(ch[index]!='\0'){
        index++;
    }
    return index;
}

// Step 02: Reverse string with two pointer approach
void reverseString(char ch[],int size){
    int s = 0;
    int e = size-1;

    while(s<=e){
        swap(ch[s],ch[e]);
        s++;
        e--;
    }
}

int main(){
    char ch[100];

    cin>>ch;

    int length = findLength(ch,100);
    reverseString(ch,length);
    cout<<"After reversing string: "<<ch<<endl;

    return 0;
}

/*
INPUT: MANOJ
OUTPUT: After reversing string: JONAM
*/
```

## 11. Program 03: Uppercase to lowercase and vice versa

lower ↓ upper

Input: My Name Is Manoj

Output: MY NAME IS MANOJ




*Lets suppose mujhe lowercase se uppercase me convert karna hai to approach kya hogi.....*

```
char low = 'c'
char uppn = low - 'a' + 'A' ;
          = 99 - 97 + 65
          = 67
```

→ C - 67 UpperCase

```
// Program 03: Uppercase to lowercase and vice versa
#include<iostream>
#include <cstring>
using namespace std;

// Lowercase to Uppercase
void convertToUppercase(char ch[],int size){
    int index = 0;

    while(ch[index]!='\0'){
        if(ch[index]>='a' && ch[index]<='z'){
            ch[index]=ch[index]-'a'+'A';
        }
        index++;
    }
}

// Uppercase to Lowercase
void convertToLowercace(char ch[],int size){
    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);

    int length = strlen(ch);
    convertToUppercase(ch,length);
    cout<<"After converting to uppercase: "<<ch<<endl;

    convertToLowercace(ch,length);
    cout<<"After converting to lowercase: "<<ch<<endl;

    return 0;
}

/*
INPUT: My Name Is Manoj
OUTPUT:
After converting to uppercase: MY NAME IS MANOJ
After converting to lowercase: my name is manoj
*/
```

T.C. = O(N) where N is length

```
After converting to lowercase: my name is manoj
*/
```

## 12. Program 04: Replace @ with the white space

Input: My@Love

Output: My Love

```cpp
// Program 04: Replace @ with the white space
#include<iostream>
#include <cstring>
using namespace std;

// Replace @ with the white space
void replaceCharacter(char ch[],int size){
    int index = 0;

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

int main(){
    char ch[100];

    cin.getline(ch,100);

    int length = strlen(ch);
    replaceCharacter(ch,length);
    cout<<"After replace to @: "<<ch<<endl;

    return 0;
}

/*
Time Complexity: O(N), where N is length of char array
Space Complexity: O(1)
*/

/*
INPUT: My@Love
OUTPUT: After replace to @: My Love
*/
```

## 13. Program 05: Check palindrome

Ex:01

Input: | B | A | B | B | A | R |

output: Invalid

Ex:02

Input: | R | A | C | E | C | A | R |

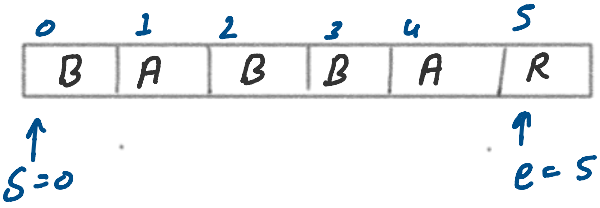
output: valid

Output: Valid

Example:01

Approach
Two pointer

Input:



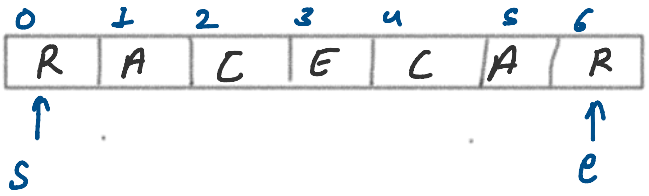

ch[S] != ch[e]
B != R
Return False

Output : Invalid

DRY RUN

Ex:02

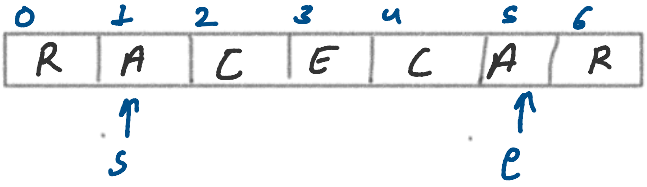
Input:




Situation:01

ch[S] == ch[e]
R == R
S++
e--

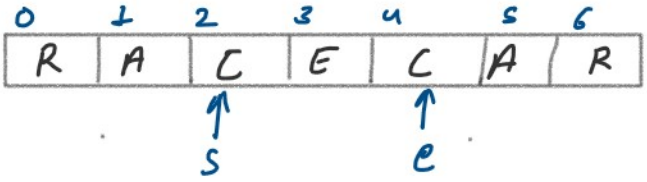
Situation:02




ch[S] == ch[e]
A == A
S++

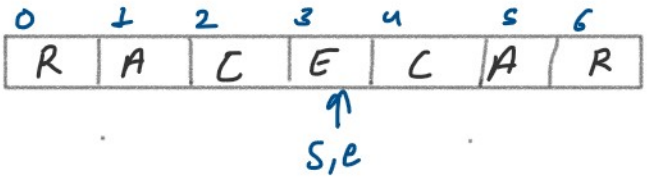
e--

Situation :03



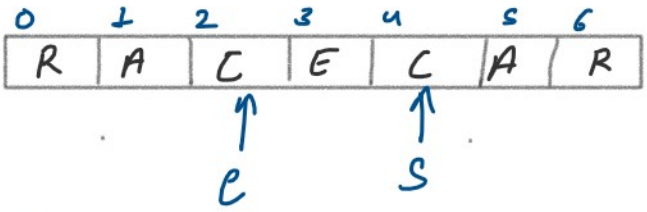

ch[s] == ch[e]
c == c
s++
e--

Situation :04




ch[s] == ch[e]
E == E
s++
e--

Situation :05




s = 4
e = 2
Ruk jaao... (s <= e)

End ✗

Output: valid

```cpp
// Program 05: Check palindrome
#include<iostream>
#include <cstring>
using namespace std;

// Check palindrome
bool checkPalindrome(char ch[],int size){
    int s = 0;
    int e = size-1;

    while(s<=e){
        if(ch[s]!=ch[e]){
            return false;
        }
        s++;
        e--;
    }
    return true;
}

int main(){
    char ch[100];

    cin.getline(ch,100);

    int length = strlen(ch);
    bool ans = checkPalindrome(ch,length);

    if(ans){
        cout<<"Valid"<<endl;
    }
    else{
        cout<<"Invalid"<<endl;
    }

    return 0;
}

/*
Time Complexity: O(N/2)=O(N), where N is length of char array
Space Complexity: O(1)
*/

/*
INPUT: RACECAR
OUTPUT: Valid

INPUT: BABBAR
OUTPUT: Invalid
*/
```

✅STRINGS

## 14. What is string?

*Strings are used for storing text. A string variable contains a collection of characters surrounded by double quotes:*

***Examples:***

1. *"Manoj"*
2. *"My Name is Love Babbar"*
3. *"123"*
4. *"My mobile number is 98971 number likh dum dika dum dika dum"*
5. *"123%%"*

```cpp
// Creation of string
string name;

// Taking input in string
cin >> name;

// Print string
cout << "Printing name: " << name << endl;

/*
INPUT: MANOJ
OUTPUT: Printing name: MANOJ

INPUTL: MANOJ KUMAR
OUTPUT: Printing name: MANOJ KUMAR
*/
```

15. Creation of string

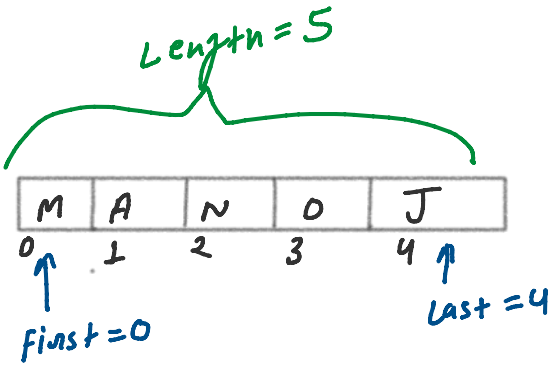
16. Taking input in string

17. Print string

## Access string by index

```cpp
// Creation of string
string name;

// Taking input in string
cin >> name;

// Access by index
cout << "Print first character: "<< name[0] << endl;
cout << "Print last character: "<< name[name.length()-1] << endl;

/*
INPUT: MANOJ
OUTPUT:
Print first character: M
Print last character: J
*/
```

⇒ 5-1 = 4




## 18. getline(cin,name) method

```cpp
// Creation of string
string name;

// Taking input in string
getline(cin, name);

// Access by index
cout << "Print first character: "<< name[0] << endl;
cout << "Print last character: "<< name[name.length()-1] << endl;

/*
INPUT: MANOJ
OUTPUT:
Print first character: M
Print last character: J
*/
```

getline (cin, name);

String variable

## 19. Char array Vs string

- String refers to a sequence of characters represented as a single data type.
- Character Array is a sequential collection of data type char.
- Strings are immutable.
- Character Arrays are mutable.
- Double quotation ("") marks are used to represent a string.
- Single quotation ('') marks are used to represent a character array.
- In a string, multiple letters or digits can be enclosed in double quotation.
- In a char array, only one letter or digit can be enclosed in single quotation.

## 20. Important predefined function of string