Week 5 - Class 2

1.
```
int main (){
    char name [100];
    cout << "Enter your Name" << endl;
    cin >> name;
    cout << "Aapka naam" << name << " hai" << endl;
    return 0;
}
```

Output:

```
Enter your Name
Aapka name: Yourname hai
```

```
char name [100];
```

100 blocks

```
cin>>name;
```

↳ by default

↳ Null character \0

↳ string termination

```
char name [100];
cin>>name;
for(int i=0; i<7; i++){
        cout <<"index:" <<i <<"value:" << name[i] <<endl;
}
```

output

index: 0 value : b
index: 1 value : a
index: 2 value : b
index: 3 value : b
index: 4 value : a
index : 5 value : r
index : 6 value : 0

After that → garbage value

Q: (A) char array create:
(B) full name store kro
(C) print

```
char arr[100];
cin >> arr;
cout << arr;
```

① Input
Love Babbar

Output
Love

② Input
Love
Babbar

Output
Love

③ Input
Love Babbar

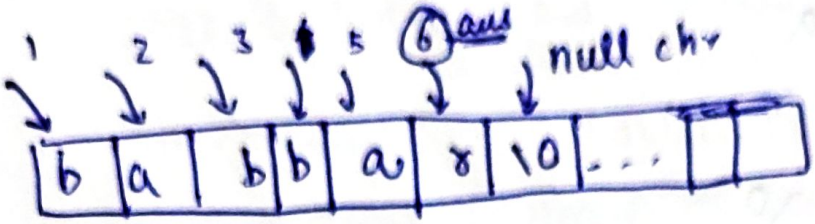
output
Love

Note: To solve this issue use 'getline()'.
`getline(cin, arr);`

- We can choose our delimator.

① Length of string

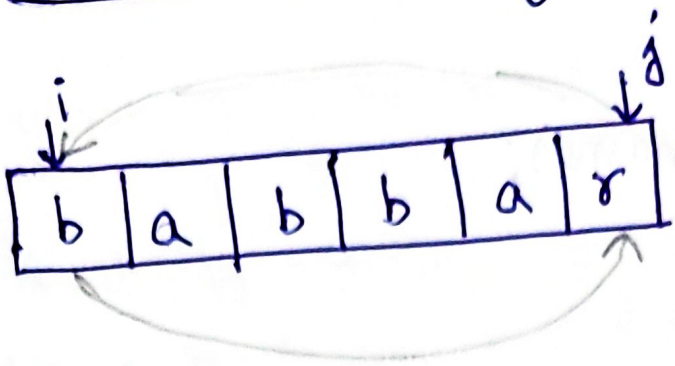
```
char name[100];
cin >> name;
```

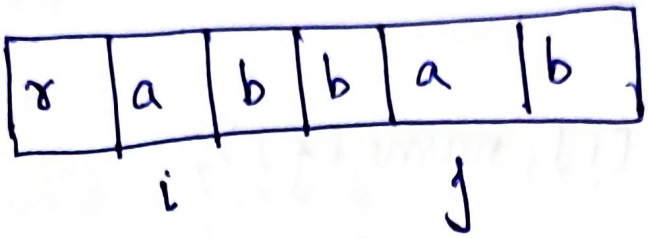


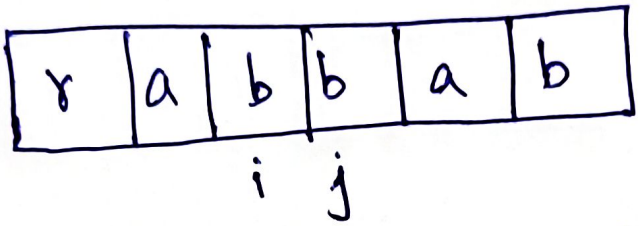

```
int getLength(char name[]){
        int length = 0;
        int i = 0;
        while(name[i] != '\0'){
                length++;
                i++;
        }
        return length
}
```

Note: strlen(name) for length of string as ho skta hai.

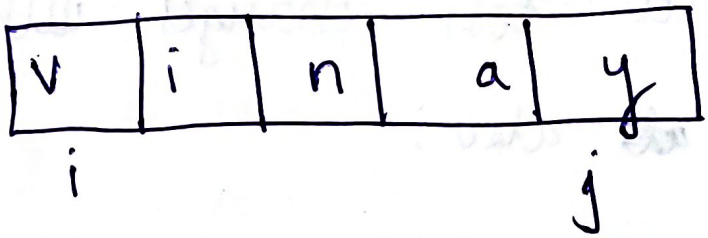
② Reverse a string :-



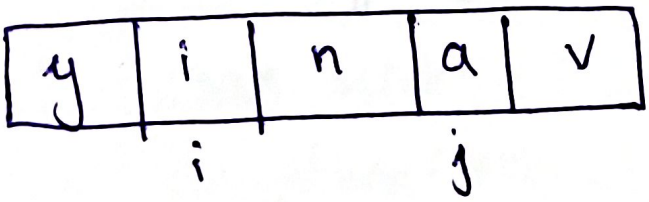


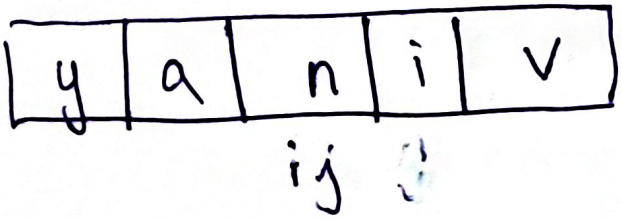


v



swap





```
void reverseCharArray(char name[]){
        int i=0
        int n=getLength(name);
        int j = n-1;
        while (i <= j){
                swap(name[i], name[j]);
                i++;
                j--;
        }
}
```

Array is pass by reference toh changes will sustain in function also.

Vector is pass by value, copy of it is created.

T.C = O(n)

S.C = O(1)

③ Replace all spaces :-

space → @

i/p → My name is Ram

o/p → My @name @ is @Ram

```
void replaceSpaces(char sentence[]){
        int i=0;
        int n = strlen(sentence);
        for(int i=0 ,i<n ; i++){
                if(sentence[i]==" "){
                        sentance[i]='@';
                }
        }
}

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

      replaceSpaces(sentance);
      cout<<"printing sentance:"<<sentence<<endl;

  return 0;
}
```

T.C: O(n)

S.C: O(1)

(9) very important Question:

Palindrome :-

→ noon

left to right → noon

right to left → noon

i/p → racecar

true false ?

1 → i/p → racecar

→ reverse → racecar

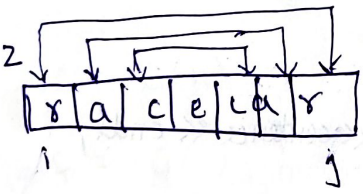
→ compare

T F ?

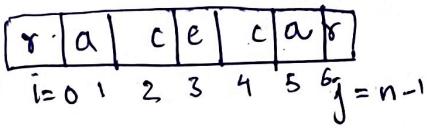
O(n) = T.C

O(n) = S.C.

2



2 pointer approach



(i = 0) == (j = n-1) ?

(i = 1) == (j = n-2) ?

(i = 2) == (j = n-3) ?

i = 3 ; i = j ✓

```
bool checkPalindrome(char word[]) {
    int i=0;
    int n= strlen(word);
    int j =n-1;
    while(i<=j){
        if(word[i]!=word[j]){
            return false
        }
        else{
            i++;
            j--;
        }
    }
    return true;
}
int main(){
    char arr[100] = "BabbaB";
    cout << checkPalindrome(arr);

  return 0;
}
```

T.C. = O(n)

S.C. = O(1)

⑤ Convert into UpperCase.

(i/p) → babbar

↓

(o/p) "BABBAR"

Code:

```
void convertIntoUpperCase(char arr[]){
    int n = strlen(arr);
    for(int i=0; i<n; i++){
        a[i] = a[i] - 'a' + 'A';
    }
}
```

→ (-65 + 97) = 32

Time complexity = O(n)

Space " = O(1)

Homework

+ a - A for lowerCase

⑥ Convert to lowerCase.

(i/p) → BABBAR (o/p) → babbar

```
void convertIntoLowerCase(char arr[]){
    int n = strleng(arr);
    for(int i=0; i<n; i++){
        a[i] = a[i] - 'A' + 'a';
    }
}
```

→ String :-

- Do not care about size.
- It is a datatype, similar to vector.

```
int main(){
    string str;
    getline(cin, str);
    cout << str;
    return 0;
}
```

Love Babbar

- str.length() → For length
- str.empty() → To check
- str.push_back('A'); → Love BabbarA
- str.pop_back(); → Love Babbar
- str.substr(0,6) << endl → Love B

Yei bhool sakte hai!

(1, 3) → ove

- if (a.compare(b) == 0);

H.W To implement compare funch.

```
bool compareString(string a, string b){
    if(a.length()!=b.length())
        return false;
    else{
        //int j=0;
        for(int i=0; i<a.length(); i++){
            if(a[i]!=b[j]){
                return false
            }
            j++
        }
    }
    return true
}
```

T.C = O(n)

```
string x = "abcd";
string y = "bcda";
cout << x.compare(y) << endl;
return 0;
}
```

O/P
-1

```
string sentence = "hello Jee kaise ho saare";
string target = "hello";
cout << sentence.find(target);
```

O/P

0

```
if (sentence.find (target) == std::string::npos){
        cout << "Not found" << endl;
}
```

↳ not found

↓

that substr not found.

- Replace()

```
string str = "This is my First Message";
string word= "Babbar";
str.replace(0, 4, word)
cout << str << endl;
```

O/P

Babbar is my First Message

- Erase():-

```
string str = "ABCDEFGHIGJ";
str.erase(0, 4);
          (10, 10)
cout << str;
```

EFGHIGJ

ABCDEFGHI