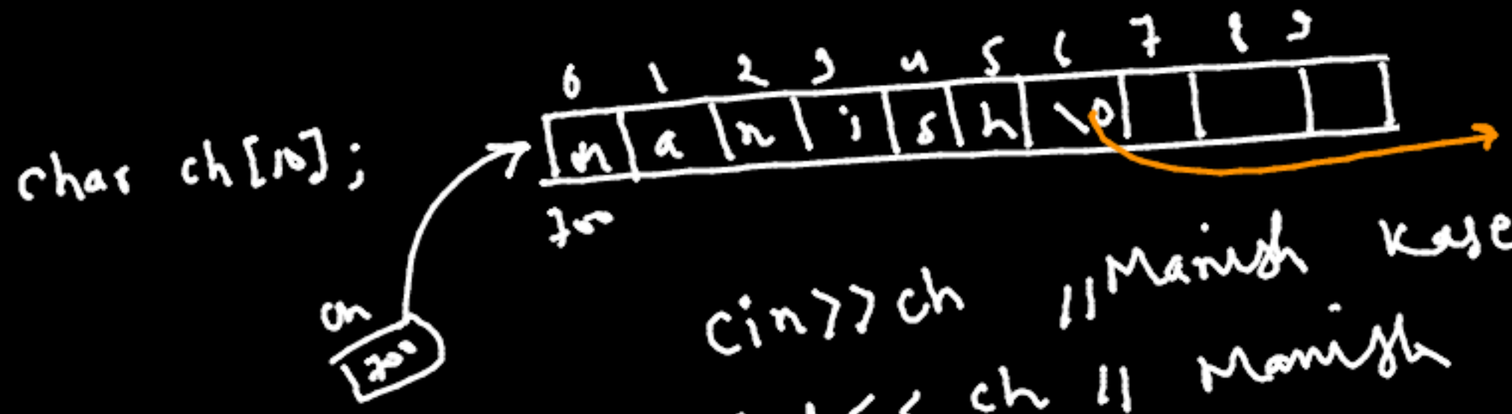


cin>>ch // Manish kayela

cout<<ch // Manish

cin.getline (ch, length, delimeter);

↑

optional

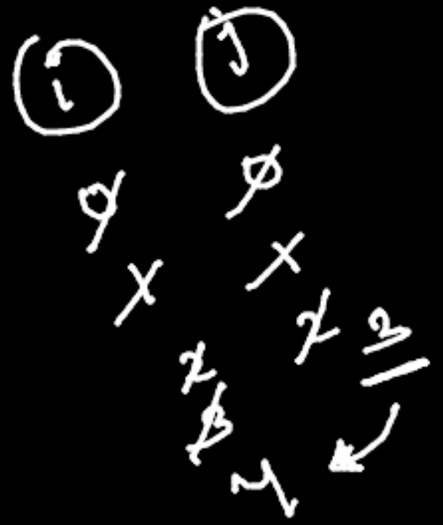
Important Questions

(i) Trim space
(ii) Reverse word wise
(iii) All substring
(iv) Wave print (Try kr skte ho)
(v) Spiral print (Imp)
(vi) Compress str.

Time Space

abc de f gh i

abcdefghi

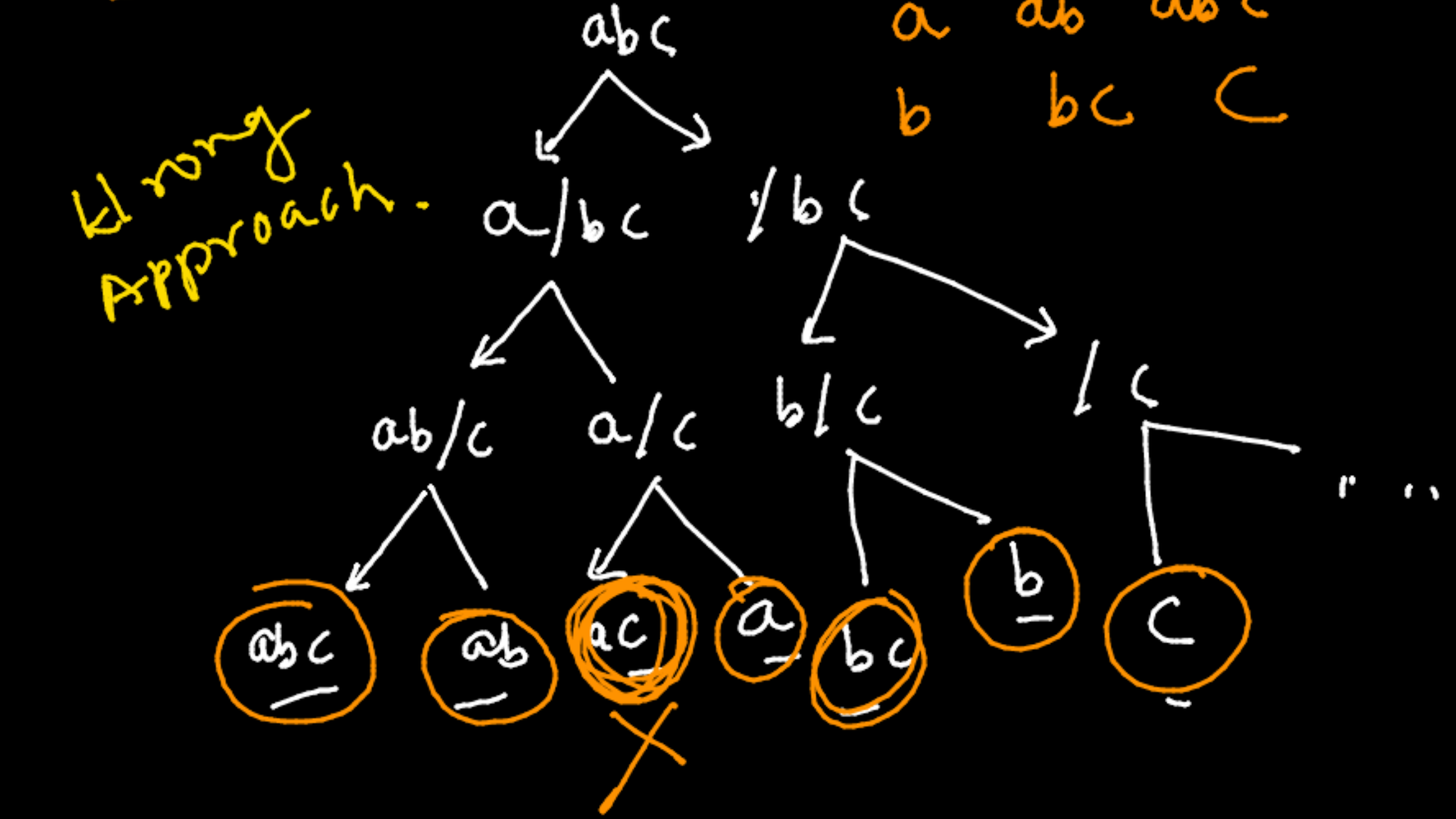



```
int j=0
for (int i=0; i<strlen(input); i++)
    if (input[i]!=' ')
        input[j] = input[i];
        j++;
input[j] = '\0';
```

All substring

abc
l
a
ab
abc
(Y) ac
b
c
bc

Klnorg Approach.



abc

a ab abc
b bc c

All substring

abcd → copy string into a string

→ a ab abc abcd
→ b bc bcd
→ c cd
→ d.

```
s;
for (int i=0; i<strlen( s ); i++)      // strlen(s) = n
{   int len=1;
    while (len <= n-i)
    {
      cout<< s.substr(i, len);
        len++;
    }
}
```

abc

i	len	3
~~0~~	~~1~~ ~~2~~ ~~3~~ 4	4 <= 3
1	~~1~~ 2 3	1 <= 2

a ab abc
b bc
c

Column Wise Sum

4 2
1 2
3 4
5 6
7 8

row col
4 2

$C_1 \Rightarrow 16$
$C_2 = 20$ } Ans

```
for (int i=0; i<col; i++)
{ int sum=0;
  for (int j=0; j<row; j++)
  {
    sum = arr[j][i]
  }
  cout<<sum<<" ";
}
```

Check Permutation

apple

```
int hash[128]={0};
for(int i=0;i<n1;i++)
    hash[str[i]]++;
for (int i=0;i<n2;i++)
    has[str[i]]--;
for (int i=0;i<128;i++)
    if(hash[i]>0)
        return false;
return true;
```

Remove Consecutive

s

aabccbaa

e

abcba

---

s → a a b c c b a a

e

a b c c b a a

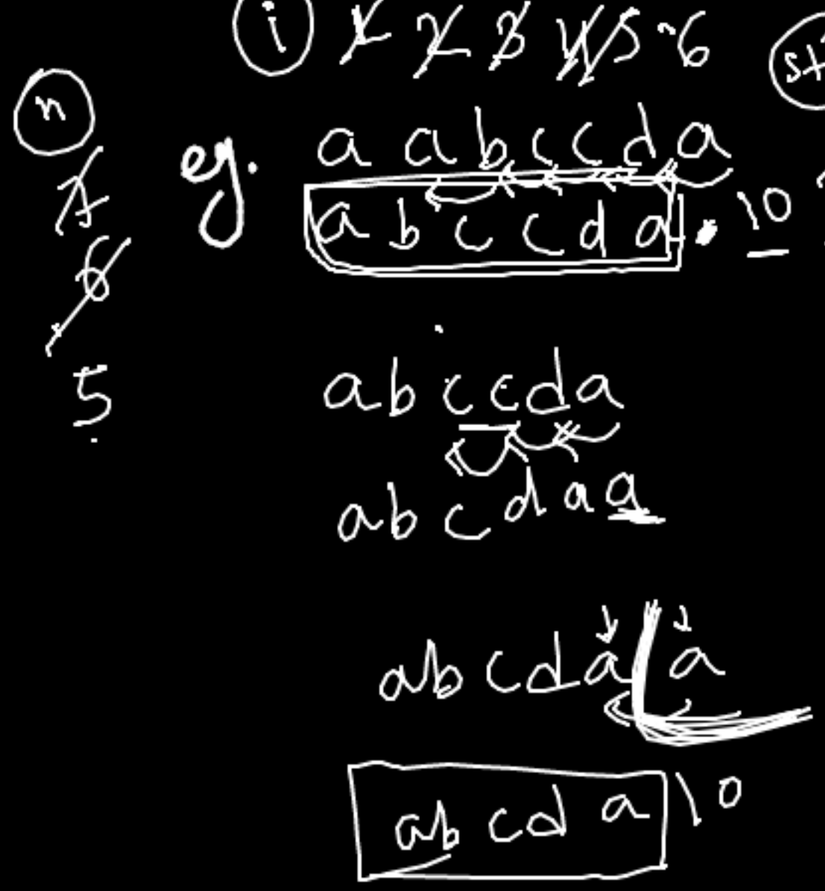
a b c b a a

abcba

```
start = 0;
end = 1;
n = strlen(str);

while( start < n && end < n )
{
    if (str[start] == str[end])
    {
        int i = end;
        for(int i=end; i<strlen(str); i++)
        {
            str[i] = str[i+1];
        }
        str[i] = '\0';
        n--;
    }
    else{
        start++;
        end++;
    }
}
```

size of char array ↓




```
int start=0, end=1, n = strlen (str);
while (start < n && end <n)
{
   if (str[start] == str[end])
   { int i=end;
     for ( ; i<n;i++)
      { str[i] = str[i+1]
      }
     str[i] = '\0';
      n--;
   }
   else {
     start++;
   } end++;
}
```

Largest Row or Column

6 9 8 5 ⇒ 28

9 2 4 1 ⇒ 16

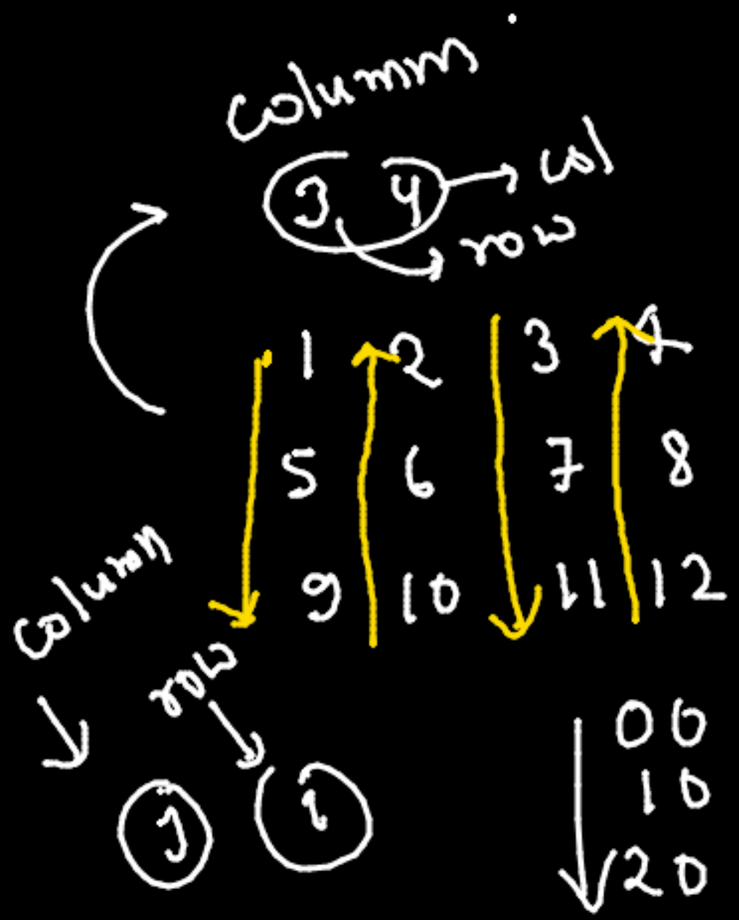
8 3 9 3 → 23

8 7 9 6 → 30

1 1 1 1

31 21 29 15

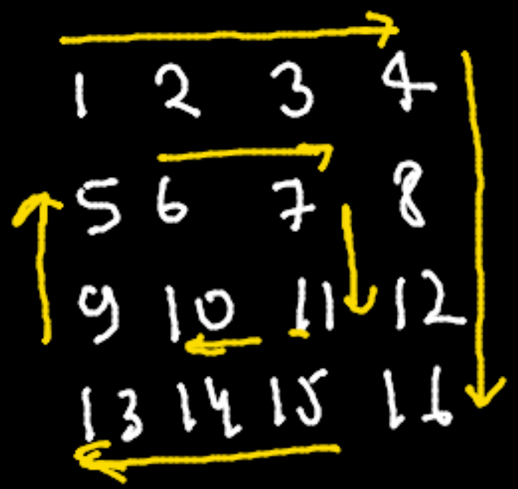
Wave Print



1 5 9 10 6 2 3 7 11 12 8 4

```
for (int j=0; j< col; j++)
{
  for (int i=0;  i<row; i++)
  {
    if (j%2==0)
      cout << arr[i][j];
    else
      cout << arr[row-i-1][j];
  }
}
```

Spiral Print (Most Imp)



1 2 3 4 8 12 16 15 14 13

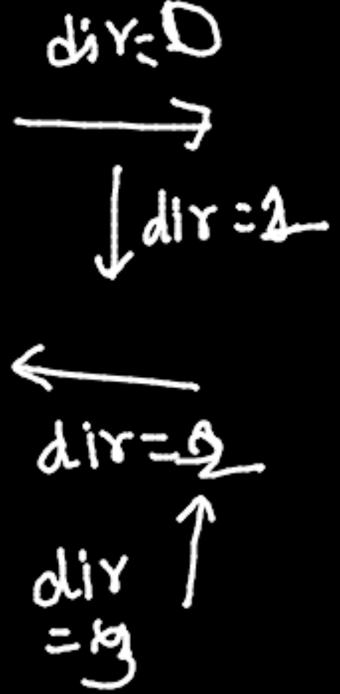
9 5 6 7 11 10

top = 0, down = row - 1

left = 0, right = col - 1;

dir = 0

↓




```
while (top <= down && left <= right)
{
    if (dir == 0)
    {
        for (int j=left; j<=right; j++)
            cout << input[top][j];
        top++
    }
else if (dir==1)
    {
        for (int i=top ; i<=dob; i++)
        {cout << input[i][right]<<"!
        } end
        right--
    }
```

```
elseif (dir == 2)
{
  for (int j = right ; j >= left ; j--)
  { cout << input [down] [j] << "  ";
  }
  down --;
}
else if (dir == 3)
{
  for (int i = down; i >= top ; i--)
      cout << input [i] [left] << "  ";

  left ++
}
dir = (dir + 1) % 4;
```

End

Compress String

eg. aaaxxppdsa

→ a3x2p2dsa

```
count=1
int start=0, end=1, string s="";
while( start<n && end<n)
{   int count=1;
    if ( input[start] == input[end] )
        count++; end++;
    else {
        s+= input[start]; → end=end+1 if(count>1)
        start=end;          s= s+to_string(count);
                count=1;
    }
```

a3x2Pd