Strings

- Array of characters
- Sequence of characters

Contest
90 min 3ques
E M H

"omansh" {'o', 'm', 'a'...} 22/23rd

How are characters stored ? ASCII

A - 65	a - 97	0 - 48
B - 66	b - 98	1 - 49
C - 67	c - 99	2 - 50
⋮	⋮	⋮
Z - 90	z - 122	9 - 57

+32 (A → a)
+32 (B → b)

Obs: Between capital & small letters, ASCII value difference is 32

```
char ch = 'b'        98

if (ch > 22) {
    print "yes"      "ak123"
}
```

Q1) Given a char s[], Toggle every character.
uppercase & small case
Toggle → capital → small
small → capital.

Eg: AnaConDa → aNAcONdA

```
for( i=0; i<N; i++) {
  if ( s[i] >= 65 && s[i] <= 90)
        s[i] = s[i] + 32
  else
        s[i] = s[i] - 32
}
return s
```

TC: O(N) SC: O(1)

array → subarray

string → substring

"bxcd"

$\frac{4 \times 5}{2}$ $\frac{20}{2}$ 10

array length $n \rightarrow \frac{n(n+1)}{2}$

string length $n \rightarrow \frac{n(n+1)}{2}$

0 1 2 3

"a m a n"

s = 1 e = 2

↳ ma

Q2 Given char s [], check palindrome lowercase alphabets.

same forward & backward

WOW	MAM	MADAM
NAMAN	MOM	LIRIL
RADAR	AMMA	MALAYALAM

Idea ⇒ What about 0^{th} & $n-1^{th}$ char?

1^{st} & $n-2^{th}$

2^{nd} & $n-3^{rd}$ all same

Check from 2 ends

M A L A Y A L A M
↑ ↑

i , j

start = 0 end = n-1

```
while (start < end) {
    if ( s[start] != s[end] )
        return false

    start++          end--
}
return true
```

TC: O(N)
SC: O(1)

M A D A M

↑ ↑

a[173] a[173]

3, 7

anamadamm

feacabacabgf

adaebcdfdcbetggte

yzannazy

Q3 Given a string. Find length of longest palindrome substring.

0 1 2 3 4 5 6 7

a b c d e f g h

s[4:6] : efg len = 6 − 4 + 1

Palindrome: String that reads the same forward & backward.

NAMAN

NAMAN

```
isPalin ( s, start, end ) {

    i = start
    j = end

    while ( i < j ) {
        if ( s[i] != s[j] )
            return false.
        i++
        j--
    }
    return true
```

TC: O(N)

```
ans = 0
for (i=0; i<n; i++) {
    for (j=i; j<n; j++) {
        if (ispalin(s, i, j))
            ans = max(ans, j-i+1)
    }
}
```

TC: $O(N^3)$

x b d y 3 3 y d b d y 3 y d x

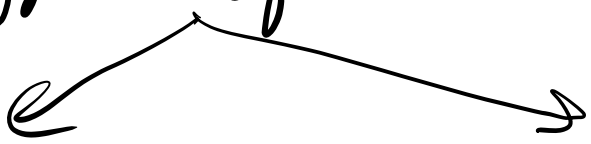
a x x a
b c y y c b

a a b b a a

a x a
p q r q p

Take all possible centers & expand.

2 types of centers



2 char i, i+1

1 char. i, i

0	1	2	3	4	5	6	7	8	9
x	b	d	y	3	3	y	d	b	d

left = 0
right = 9

What is actual length?

palin left+1 1
right-1 8

[left+1, right-1)

b - a + 1

right - left - 1

r - l + 1 - 2 = r - l - 1

```
int extend (char s[], int i, int j){
    while (s[i] == s[j] && i >= 0
                        && j < n)
        i--;        j++;
    return j - i - 1;
}
```

0	1	2	3	4	5	6	7	8	9
x	b	d	y	3	3	y	d	b	d

↑ ↑

0	1	2	3	4	5
b	a	b	a	b	c

```
int largest_pal_substring(char s[]){
    int ans = 0
    for(i=0; i<n; i++){
        // take s[i] as center.
        len = expand(s, i, i)
        ans = max(ans, len)
    }
```

single char center

```
    for(i=0; i<n-1; i++){
        // take s[i], s[i+1] as centers.
        len = expand(s, i, i+1)
        ans = max(ans, len)
    }

    return ans.
```

TC: $O(N^2)$ SC: $O(1)$

Java Specific

String: Cannot change {Immutable}

char s[] : Can change any index.

String Builder : Can change any index
class in Strings

```
String name = "omansh"

name[0] = 'O'
```