SCALER

String Basics

TABLE OF CONTENTS

String

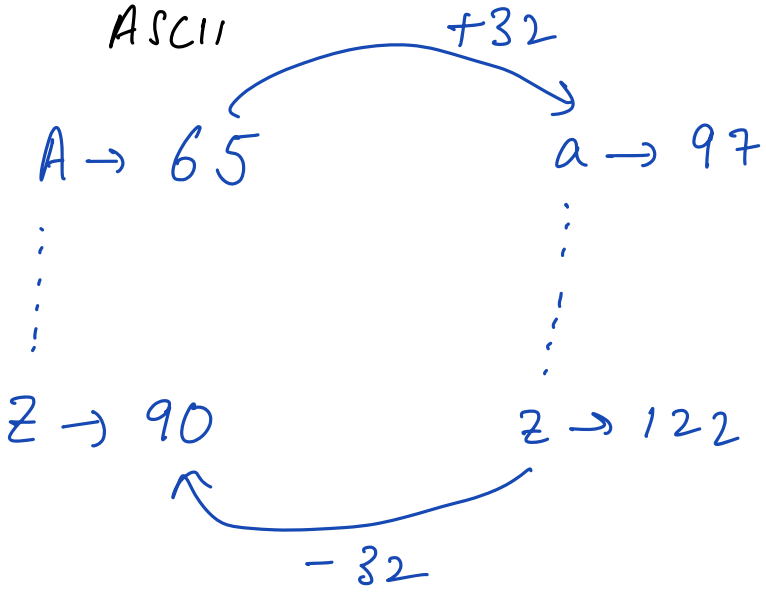
"Hello World"

sequence of characters

Character

'A' 'C' 'e' 'z' '9'

symbols that represent alphabet set



1. **char ch = 'a';**

   print(ch); a

2. **int ch = 'a';**

   print(ch); 97

3. **int ch = 'a' + 1;**

   print(ch); 98

Switch Case

< *Question* > : Given a string consisting of lower-case and upper-case alphabets.

Convert: (1) lowercase → uppercase

(2) uppercase → lowercase

1. **"H e l l o"** - hELLO

2. **"a D g b H J e"** - AdGBhjE

68

< / > Code

```
function solve ( string s) {
    N = s.length()          len(s)
    for( i : 0 → N-1 ){
        if ( s[i] >= 65 && s[i] <= 90){
            s[i] = s[i] + 32
        }
        else {
            s[i] = s[i] - 32
        }
    }
}
```

Substring

1. Contiguous part of a string.
2. A single character is also a substring.
3. Whole string is also a substring.
4. Empty string (" ") is not a substring.
5. String of length N. How many substrings will be there? → $\frac{N(N+1)}{2}$

abcd N = 4 $\frac{4(5)}{2}$ $\frac{20}{2}$ 10

raparar

madam

racecar

Check for substring if it's a palindrome or not.

madam
mom
racecar

si ei

str - a , b , m , a , d , a , m , t , a , m si = 2 , ei = 6

0 1 2 3 4 5 6 7 8. 9

true else false

```
boolean isPalindrome( String str, int si, int ei ) {
    i = si          j = ei
    while ( i < j ) {
        if ( str[i] != str[j] ) {
            return false
        }
        else { i++  j-- }
    }
    return true
}
```

adaebcdfdcbe tjgte

len = 9

< Question > : Given a string s. Find the length of the longest palindrome substring in s.

s → "a n m a d a m m"

$1 \le N \le 10^3$

ans = 5

f e a c a b a c a b g f

ans = 7

$i - j + 1$

$[i, j]$

$\Rightarrow j - i + 1$

[BF Idea] - Iterate over ALL substrings.

Brute Force idea

```
ans = 0
for ( i : 0 → N-1 ) {
    for ( j : i → N-1 ) {
        if ( isPalindrome (s, i, j) == true)
            ans = max (ans, j - i + 1)
    }
}
```

TC? $O(N^3)$

madam

daad 2, 12
3, 11
⇒ 9

[Idea - 2] - "a, d, a, e, b, c, d, f, d, c, b, e, t, g, g, t, e"
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

odd → 1 char center even → 2 char centers

Technique ⟹ expansion

```
ans = 0
for( i : 0 → N-1 ){
    // center is idx i
    left = i        right = i
    while( left >= 0 && right < N ){
        if( s[left] != s[right] )
            break
        else {
            left --    right ++
        }
    }
    actual_start = left + 1
    actual_end = right - 1
    len = actual_end - actual_start + 1
    ans = max(ans, len)
}
```

< / > Code

```
for( i : 0 → N-1) {
    // center is idx  i, i+1
    left = i          right = i+1
    while( left >= 0 && right < N ){
        if( s[left] != s[right])
            break
        else {
            left --    right++
        }
    }
    actual_start = left + 1
    actual_end = right - 1
    len = actual-end - actual-start + 1
    ans = max(ans, len)
}
```

TC: $O(N^2)$

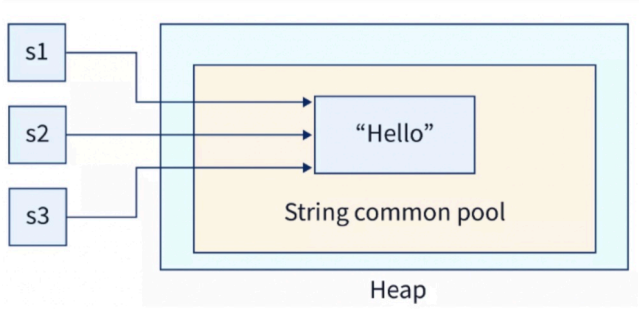
Immutability of the strings

aello

String s1 → "H e l l o"

String $s_2 = s$

$s_1[0] = 'a'$



String Builder ⇒ allows you to add & change characters