Strings:

- Intro , ASCII codes
- Flip
- sort char array
- Longest Palindromic Substring

Strings def. : array of chars.

characters : ASCII values   Google ASCII table

a-z
26

0 'A' → 65 -32 ⟷ +32 'a' → 97   '0' → 48

1 'B' → 66   'b' → 98   '1' → 49

2 'C' → 67   ⋮   ⋮

ch-65   ⋮   '9' → 57

⊛
ch-'A' 25 'Z' → 90 -32 ⟷ +32 'z' → 122   '10' → x 49 48
↳ index   ↘ not single char

C/C++   ~~UTF-8~~

8 bit 1 byte   ~~UTF-16~~ ...

Java	Python
PrintLn((int)'A') 65	print(ord('A')) 65
PrintLn((char) 66) B	print(chr(66)) B
PrintLn('B'>'A') true	print('B'>'A') True
PrintLn('C'<'A') false	print('c'<'A') false
PrintLn('C'-'A') 2	print(ord('C')-ord('A')) 2

$\overbrace{\text{'a'-'A'}}^{32}$

ex string s= "abcd"
$\quad\quad\quad\quad$ 0 1 2 3

index print(S[0]) → a   s.charAt(0)

print(S[2]) → c

P1 Given a char array of only alphabet chars
toggle every char Capital ⇄ Small

no built in functions

'a' ... 'z'
'A' ... 'Z'

ex AnaConDa
aNAcONdA

Quiz*
TC: O(n)
SC: O(1)

```
char[] toggle(char s[]){
    int n = s.len
    for(i=0; i<n; i++){
        if('a'<=s[i] && s[i]<='z'){
            s[i] = (char)(s[i] - 32) → ('a'-'A')
        }else{ // capital
            s[i] = (char) (s[i] + 32 )
        }                   → ('a'-'A')
    }
    ret s
}
```

32
0010 0000

Part 2:
optional
use XOR
&
Bitmanipulation

'A' →	65	0100 0001 7654 3210	'a' →	97	0110	0001
'B' →	66	0100 0010 7654 3210	'b' →	98	0110	0010
⋮		⋮	⋮			
'Z' →	90	0101 1010	'z' →	122	0111	1010

1 ^ 1 = 0
0 ^ 1 = 1

$s[i] = s[i] \text{ ^ } \underbrace{32}_{(1<<5)}$ ← solution one line

P2 Given a char, array s[], that contains only lower case alphabet letters, sort S[] chars in alphabetic order

no of chars in arr ← $1 <= n <= 10^5$

'a' <= ch[i] <= 'z'

ex ch[] : { d , a , b , a , c , d , z , d }

a , a , b , c , d , d , d , z

idea1: Arrays.Sort(ch) ✓

TC: nlog(n)

SC: o(1)

exist[ch - 'a']

i

ch[] : { d , a , b , a , c , d , z , d }

⚠ idea2: {a,a,b,c,d,d,d,z}

freq count

freq[index]++

	char	count
0	'a'	~~0~~ ~~1~~ 2
1	'b'	~~0~~ 1
2	'c'	~~0~~ 1
3	'd'	~~0~~ ~~1~~ ~~2~~ 3 Ⓙ
4	'e'	0
	⋮	
25	'z'	~~0~~ 1

{a,a,b,c,d,d,d,z}

i

k

Quiz

TC:O(n)
SC:O(1)

```
void  sortString(char s[]){
    int n = s.Len
    int freq[26] = {0}
    for (i=0; i<n; i++){
n       index = s[i] - 'a'
        freq[index]+=1
    }
    k=0
    for(i=0; i<26; i++)
        char ch = i+'a'
n       for( j=0; j<freq[i]; j++){
            s[k]=ch;  k++;
        }
    }
    ret s;
}
```

O(n+n)

Substring ⟶ Subarray

1. Continouse part of string
2. Full string is substring
3. Single char can be substring

how to identify a substring?

s,e → start index, end index

true/false

P3 Check if given subString is palindrom or not?

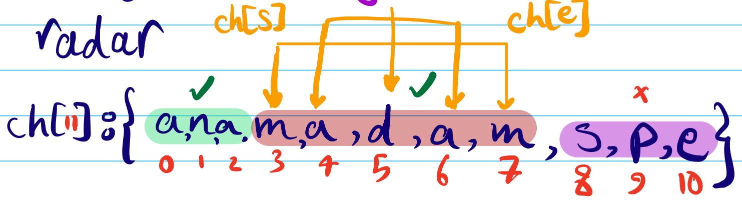
ex

mom civic "x" → yes. is palindrom

dad level

madam radar

x y y x



```
bool isPalindrom(char ch[], int s, int e){
    while(s<e){
        if( ch[s] != ch[e]) ret false;
        else{ // ch[s] == ch[e]
            s++
            e--
        }
    }
    ret true
}
```

TC: O(n)

SC: O(1)

$\frac{e-s}{2} \rightarrow \frac{n}{2}$

output int

P4 Given string, calculate length of longest Palindrom substring

Quiz* a b a c a b
0 1 2 3 4 5
ans=5

Quiz* a b c d e
0 1 2 3 4
ans=1

Quiz*

TC: O(n^3)
SC: O(1)

```
int longestPalindrom1( char ch[]){
    int n=ch.Len    ans=0
    int ans = 0;
    for(i=0; i<n; i++){
        for( j= i ; j<n; j++){
            if( isPalindrom(ch, i, j)){  //O(n)
                ans = Max(ans, j- i+1)     [a,b]
            }                   a-b+1
        }
    }
    ret ans
}
```

idea

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ex	x	b	d	y	z	z	y	d	b	d	y	z	y	d	x
					↑								↑		

Case 1: pallindrom has odd length with the center given $i \genfrac{}{}{0pt}{}{\nearrow s}{\searrow e}$ $\boxed{--}\ \underset{\downarrow}{s}\ \underset{\downarrow}{e}\ \boxed{++}$ madam

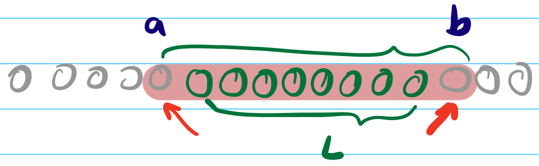
strech as much as stay pallindrom

Case 2: pallindrom has even length with the center pair is given (i, i+1)

strech as much as stay pallindrom

$O(n)$

```
int expand(char S[], int P1, int P2){
    while( P1>=0 && P2<s.Len && S[P1] == S[P2])
        P1-- ; P2++ ;
    }
    ret P2-P1-1
}        b-a+1
```



$L2 - 2 = L$

$P2 - P1 + 1 - 2 = P2 - P1 - 1$

Quiz*

TC: $O(n^2)$

SC: $O(1)$

```
int longestPalindrom2(char ch[]){
    n = ch.len
    ans = 1
    for(i=0; i<n; i++){   //odd case ①
  n     P1 = i ; P2 = i
        ans = max(ans, expand(ch, P1, P2))
    }
    for(i=0; i<n-1; i++){   //even case ②
        P1 = i;   P2 = i+1;
        ans = max(ans, expand(ch, P1, P2))
    }
    ret ans
}
```

$O(n^2)$

$O(n^2)$