collection / sequence / array of characters
text

abac

cat → tac

String → Binary form

ASCII codes

'0' → 48
'1' → 49
'2' → 50
'a' → 97

'A' → 65
'B' → 66
'C' → 67
'D' → 68

'a'-'A' → 32

'z'-'a' → 25

'z' → 122

'a' → 97

'a'-'z' → 90

"aloe"

cat → 97 99 116
c a t

hatt

h a t r d t

char

string s = "hatt";
cout << s[0] << endl; → h
cout << (int)(s[3]) << endl; → 112
s[1] ⇒ 1th character.
"cat" ⇒ s.toLowerCase();

Q.1) Given a string s, toggle the case of every character.

s → AbaEFgH
AbDefGh

fn(i:=0; i<n; i++){
if(ch[i]>='a' && ch[i]<='z')
ch[i]-= ('a'-'A');
else if(ch[i]>='A' && ch[i]<='Z')
ch[i]+= ('a'-'A');
}

ch[i] ^ (1<<5)

65 = 64+1 = 'a'-'A'

(90)₁₀ = (1011010)₂

32 = 2^5 = 100000

65 → 1000001

'A' to 'Z' → 0

10
1 0 0 0 0 0
32 → 1
11

'a' to 'Z'

'a' → 1 0 1 0 0 0 0 1
'A' → 1 0 0 0 0 0 1

1<<5

ch[i] ^ (1<<5);

Time → O(N)
S.C. → O(1)

fn(char ch = 'a'; ch <= 'z'; ch++){

}

Q.2) Given a string of lower case letters, sort the string.

→ 2.6

S → "dabacdb"
→ aabbdc
initialize int → O(nlog^n)

Array with atmost 26 unique values.

a → 2
b → 2
c → 0
d → 2
e → 1

aabbdc

97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122

a b c d e f g h i j k l m n o p q r s t u v w x y z

'a'

a → 1 [97]++
b → 1 [98]++

s[3] = 2 ⇒ 2 'a's are present

ch[i] ⇒ s[ch[i]-'a']++;

Code:

int count[26] = {0};
for(i:=0; i<n; i++){
count[s[i]-'a']++;
}
string s;
for(i:=0; i<26; i++){
while (s[i]>0){
s.append((char)(i+'a'));
s[i]--;
}
}
count[0] + count[1] + ... count[25]
= n.

O(n)

n

Counting sort

Given a string, a start index (s), an end index (e), reverse the substring b/w s and e.

e → 5
eg: a b d c a g f
s = 2
e = 5

a b g a c d f

a b d c a g f
s s+1 e e-1

reverse

reverse(char[], int s, int e){
while (s<e){
char ch1 = ch[s];
ch[s] = ch[e];
ch[e] = ch1;
s++;
e--;
}
}

a b c d e f g
s s+1 e e-1

a b c d e f g
s s+1 e

a b c d e f g
s s+1 e

a b c d e f g
s s+1 e