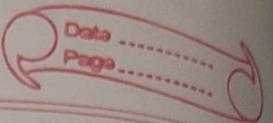


Null char does consume memory.



String

- * ASCII code - 0 to 127

A - 65	9 - 97	0 - 48
:	:	:
Z - 90	2 - 122	9 - 57

Every ASCII code takes 1 byte

- * char declaration → char temp;
- char initialization → char temp = 'A';
- to display char %c is used as format specifier

Array for char → char x[] = {'A', 'B', 'C'};

- * To show string has ended Null char '\0' is used at last index

char name[] = {'J', 'O', 'h', '\n', '\0'};
char name[] = "John";

To display string %s is used to as format specifier.

- * Now to take string value until Enter is type gets() is used.

gets(name);

Remember
char is always 'a' and not
"a"

45

Date
Page

- * Find length of string.
to find length Null character \0 needs to be found.

* changing cases

upper case \rightarrow to lower case

ASCII difference = +32, i.e. $\begin{matrix} \text{upper} - \text{lower} = 32 \\ \text{case} \quad \text{case} \\ \text{ASCII} \quad \text{ASCII} \end{matrix}$

$\begin{matrix} \text{lower} - \text{upper} = -32 \\ \text{case} \quad \text{case} \end{matrix}$

to convert from lower to upper subtract value by +32

* Vowel

Simply test if its an vowel with if statement

* Count words

- $\text{\circledast Hello World}$
- So words are mostly no of spaces + 1
- But Exception of white space - Hello world
So here we count the no of space
only if the previous index element is not space.

* Validate string

check if its character / check string at every index - If its char/number Ascii value then its fine or else string is not valid

* Reverse string

→ Method 1:

- Using another string
- copy last element of string to first position of helper array.
- After copying all element then copy reversed ~~helper string~~ element as it is to main string

• Method 2

swap first element with last element and so on

* Comparing string.

Well run the loop until one of the string ends or they are unequal then check which one is smaller or greater

* Palindrome - Reverse it and if both are same then its palindrome eg. madam, sas

→ Using another string reverse copy and check if its equal

→ Compare first element with last if so and so on if all were equal then its a

logical operator works on booleans

Bitwise operator works on bits

Date _____
Page _____

46

* Finding duplicates in a String \rightarrow using for loop
* using bitwise \rightarrow using Hashing

1 Byte \rightarrow $H = \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{1}$

$H = 1$

left shift $\rightarrow H = H \ll 2 \rightarrow \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{0}$

$a = 10 \rightarrow 1010$

$b = 6 \rightarrow 0110$

$a \& b = 0010$

$a \rightarrow 10 \rightarrow 1010$

$b \rightarrow 6 \rightarrow 0110$

$a \& b = 1110$

* Checking if a bit is on is known as Masking.

$H = \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{0}$

Check if bit 3 is on

$\therefore A = \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{0}$

Now left shift by 3

$a = a \ll 3$

Now if $H \& A$ is 0 then bit is off else on.

* Merging \rightarrow To set a bit on is known as merging

$H = \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{0}$

Setting bit 4 as on

$A = \boxed{0} \boxed{0} \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{0} \boxed{0}$ $a = a \ll 4$

Now $H = H | A$

* Q) Checking duplicates using bitwise operators
H = "Hello";

\Rightarrow Now $\rightarrow \text{int} \rightarrow 4 \text{ bytes} = 4 \times 8 \text{ bits} = 32 \text{ bits}$

No of alphabets = 26

Now if at

int bits can be used as hash tab.

→ Now

- assign int as A=0

Now using hashing method

```
for (int i=0 ; H[i] != '0' ; i++)
```

三 01112 1010

int B=1; // it means assigning & turning on the first bit

$$B = B \ll (H[i] - 97)$$

Now masking if bit is on, on that

in particular bit

if (C & D) f731 won't

cout << Duplicated

~~little~~ else / does turning on that Bit

$$H \doteq HIB_j$$

3

~~is awarded in no field other than~~

$$P \gg P = D \quad \boxed{1/0/0/3/3/0/0/3} \rightarrow A$$

ATT-11-200

- * Anagram \rightarrow same set of chars to form two different string is anagram
eg:- decimal & medical.

to find
~~OCCUR~~ Now, Anagram

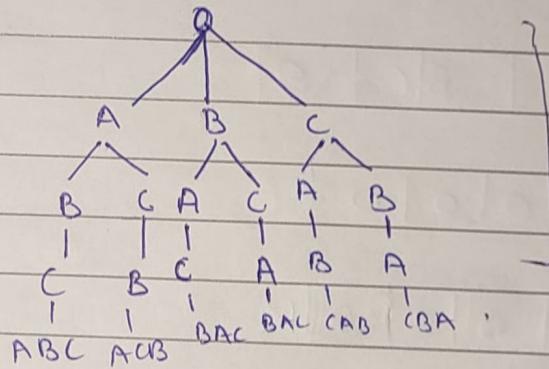
- Method 1 using for loop $\rightarrow O(n^2)$
- Method 2 using hash table
- Method 3 using bitwise operator

* Permutation of String

~~S T A B C~~

- ① ABC
- ② ACB
- ③ BAC
- ④ BCA
- ⑤ CAB
- ⑥ CBA

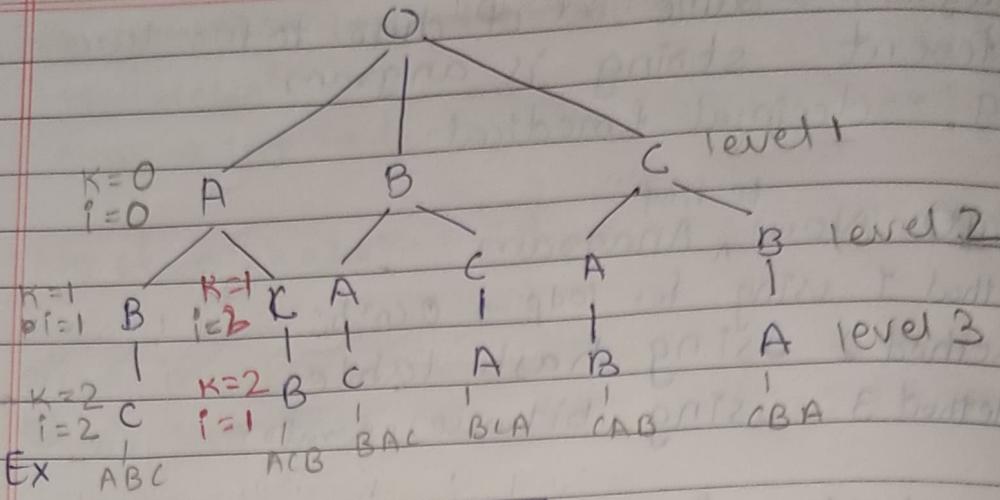
* 



This is backtracking

As we founded all possible permutation is known as Brute force -

Backtracking is used for brute force (all solution)



•) Firstly S [A | B | C | \o]

A  1

Res	A	B	C	10	K-point
	0	1	2	3	

first input Alphabet if A is available.

$$j = 0$$

二〇

S | A | B | C | \theta |

四 () 8 10 11 12

\rightarrow for $K=0$

point which corresponds to A i.e S[i]

A 110100 $i=1$

D 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

1 for $x = 1$

9-1 which correspond to Brie S[i]

A [1|1|0|0] i+=1

→ For

$K = 2$
 $i = 2$

since $A[2] = 0$

$A[1] \mid 1 \mid 1 \mid 10 \quad i+1$

$B[A \mid B \mid C \mid 10 \quad K+1]$

Now $S[K]$ equals to null character and
also $S[i]$ equals to null

→ Point R

→ Go to level 2 at adjust K & i value accordingly

→ For $K = 2$

$i = 2$

Now Here $A[i] \neq 0$

∴ increment value of i after assigning $A[2] = 0$
again $S[i] = '10'$

∴ we have to Go to level 2.

~~KA[i] = 10 → B[i] = 0~~

→ For $K = 1$

$i = 1$

Now as $A[i] \neq 0$ increment after assigning

$A[1] \mid 0 \mid 10$

∴ $i = 2$

Here $A[2] = 0$

∴ Assign ~~R~~ $R[S[R]] = C$

$R \mid A \mid C \mid 10 \rightarrow A[1] \mid 0 \mid 10$

$i+1 \quad (0 = S[1])$

$K+1$

→ For $k=2$ $A[i]=0$ at $i=1$
 Check Available A through entire string

$A[i] = A[i]=0$ for $i=1$

∴

$A[1][1][0]$

$Res[k] = B$

$R[A[C[B]\backslash\theta]$

→ $k++$

as k reached null

- Point Res

∴ Go Back At Level 3

* void perm(char s[], int k)

{ static int A[10]={0};

static char Res[10];

int i;

for (if ($S[K] \neq '\backslash\theta'$)

{ $Res[k] = '0'$; printf(Res);

}

for ($i=0, s[i] \neq '\backslash\theta'$); $i++$)

{

if ($A[i] \neq 0$)

{ $Res[k] = s[i]$

$A[i] = 1$;

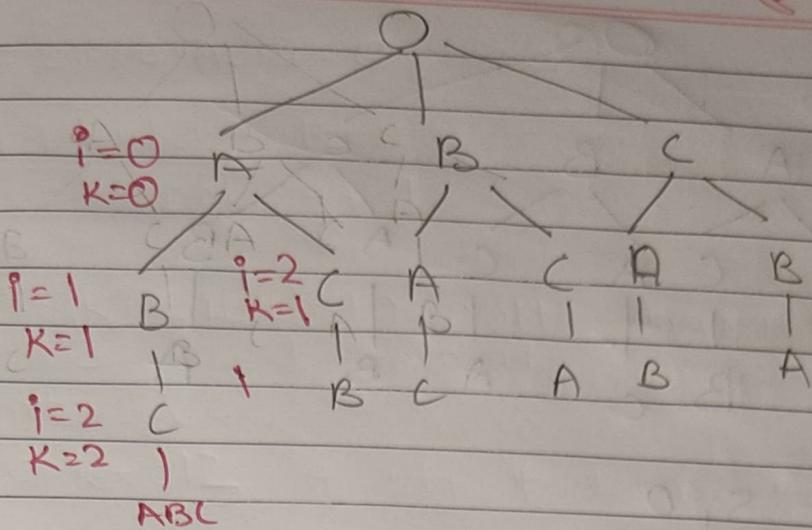
perm ($s, k+1$);

$A[i] = 0$;

}

3

3



$\rightarrow \boxed{0|0|0|0|0}$

\rightarrow For $k=2$

$i=1$

Now $A[i] = 0 \therefore$ increment i after assign $A[i]=0$

$\therefore A \boxed{1|0|1|0|0} \therefore$ again increment but don't assign
incrementing i goes to null

$A \boxed{1|0|1|0|0}$

\therefore go back to level 2

\rightarrow For $k=1$

$i=2$

as $A[i]=1 \neq 0$

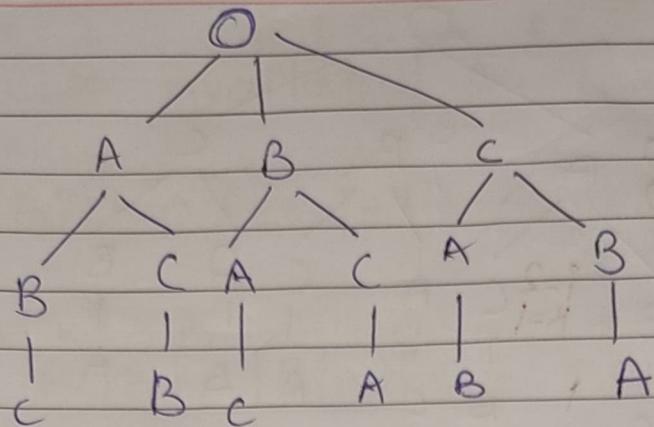
increment $i \because$ its null character go back to level 1

$A \boxed{1|1|0|1|0|0}$

\rightarrow For $k=0$

$i=1$

$\therefore A \rightarrow 1$



→ Main ← S, O

- A [0 0 0] ;
- R [] K

for (each term of S)

↓
R as Available

R [A] []

• Perm

↓

K=1 R [A] [B]

↓

K=2 R [A] [B] [C]

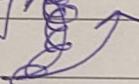
↓

K=3 Point ABC

K=1 A [1 0 0] → A [1 0 1]

↑

K=2 A [1 1 0]



A [1 1 1]

↓

R [A] [C] [B]

↓

Point A(C(B))

A [1 1 0 1]
R [A] [D] [B]

↑

* Method 2
~~S T A B | C | T o l~~

using swap

