DAY 5:

Session 1:

Alphabet Count

ID:11078 Solved By 950 Users

The program must accept a string **S** containing only alphabets as the input. The program must print the alphabets in S along with their number of occurrences in alphabetical order.
Note: The order of the output must be upper case alphabets followed by lower case alphabets.

Boundary Condition(s):
2 <= Length of S <= 10^5

Input Format:
The first line contains S.

Output Format:
The first line contains the alphabets in S along with their number of occurrences in alphabetical order.

Example Input/Output 1:
Input:
award

Output:
a2d1r1w1

Example Input/Output 2:
Input:
IndianCricketCouncil

Output:
C2I1a1c2d1e1i3k1l1n3o1r1t1u1

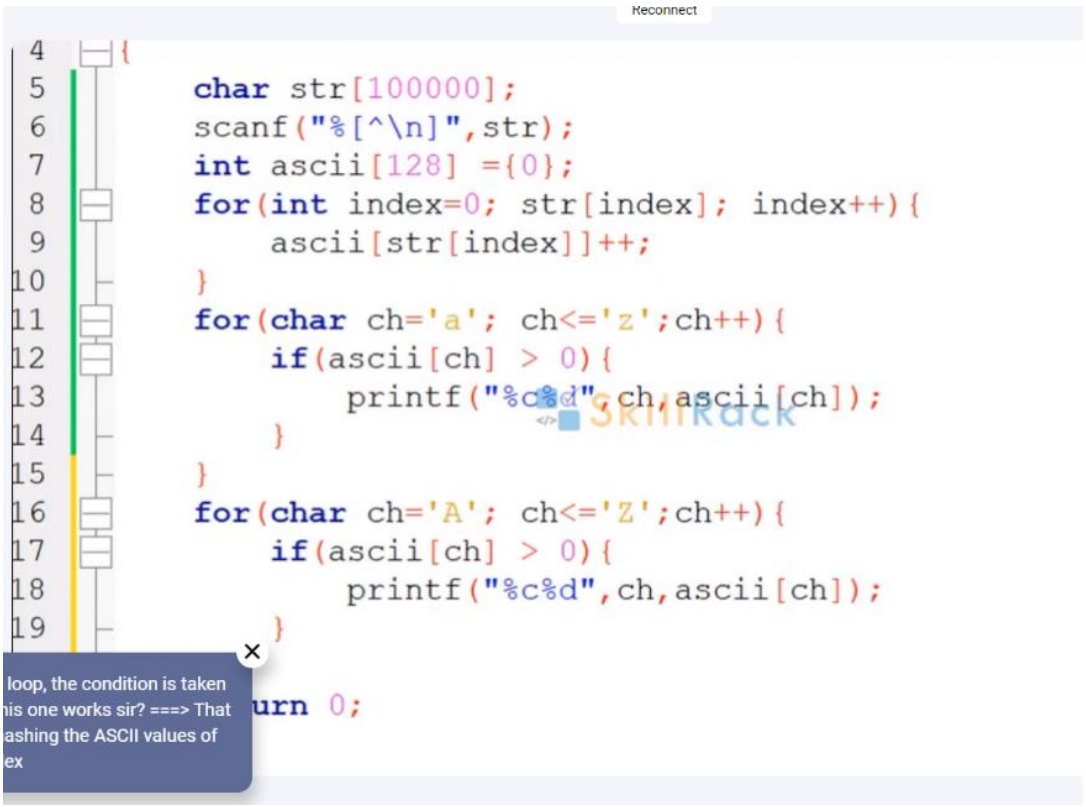
Max Execution Time Limit: 500 millisecs

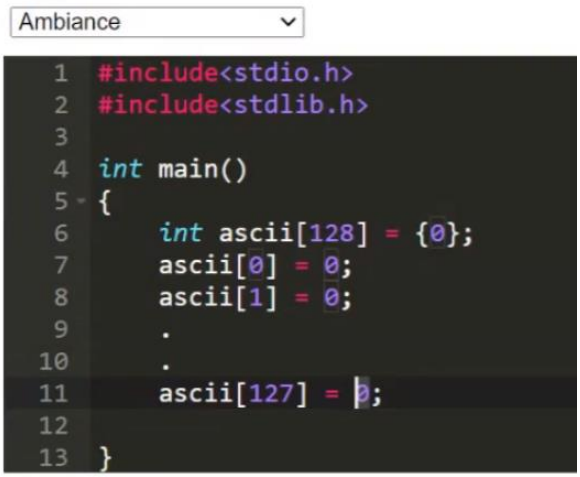
Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    char str[100000];
    scanf("%s",&str);
    int ascii[128]={0};
    for(int i=0;i<strlen(str);i++){ //i<strlen(str) can also be coded as str[i] -
->loops until the null character
        ascii[str[i]]++;
    }
    for(int i=0;i<128;i++){
        if(ascii[i]!=0){
```

```
            printf("%c%d",i,ascii[i]);
        }
    }
}
```

Alternate approach:





Ambiance

```c
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int N;
    scanf("%d", &N);
    int arr[N] = {0};

}
```

Compile error – since the N is dynamic

Hashing does not work for dynamic memory allocation

So, do this

Ambiance

```c
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int N;
    scanf("%d", &N);
    int arr[N];
    for(int index = 0; index < N; index++)
    {
        arr[index] = 0;
    }

}
```

```
    int N;
    scanf("%d", &N);
    int arr[N] = {0};
    return 0;
}
```

Your Input

10

Your Program Output:

```
Hello.c: In function 'main':
Hello.c:9:5: error: variable-sized object may not be initialized
   int arr[N] = {0};
   ^~~
```

Problem 2:

String First Repeating Character

ID:11079 **Solved By 947 Users**

The program must accept a string S as the input. The program must print the first repeating character in S as the output.

Boundary Condition(s):
2 <= Length of S <= 10^5

Input Format:
The first line contains S.

Output Format:
The first line contains the first repeating character in S.

Example Input/Output 1:
Input:
engine

Output:
n

Example Input/Output 2:
Input:
cool

Output:
o

Max Execution Time Limit: 100 millisecs

Ambiance

```c
#include<stdio.h>
#include<stdlib.h>

int main()
{
    char str[100];
    str[0] = 'a';
    str[1] = 'p';
    str[2] = 'p';
    str[3] = 'l';
    str[4] = 'e';
    str[5] = '\0';
    printf("%s", str);
}
```

specify null character at the end

Before increment code:

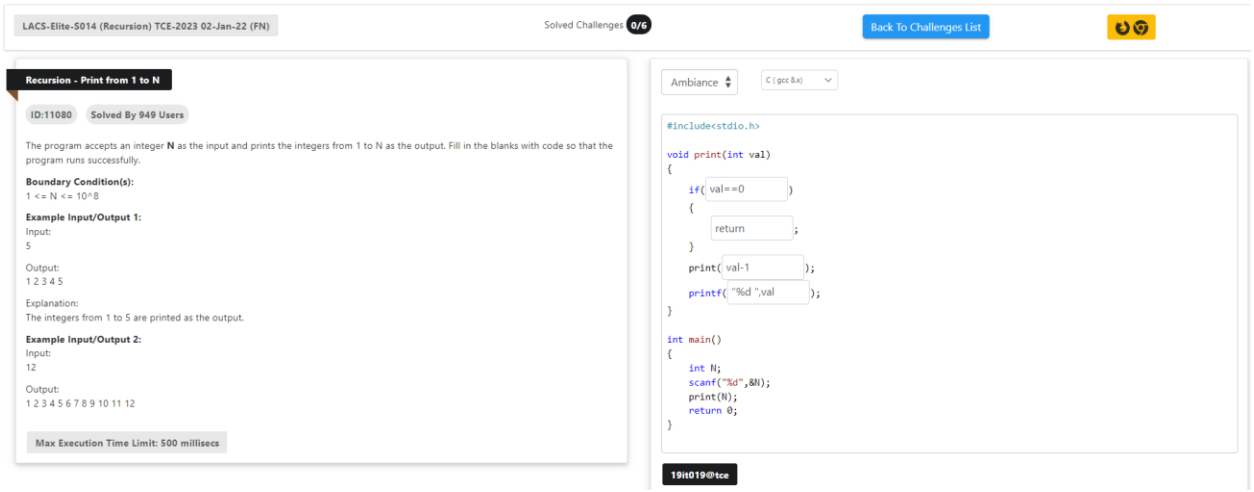
Ambiance

```c
#include<stdio.h>

int main()
{
    char str[100001];
    scanf("%s",str);
    int ascii[128]= {0};
    for(int index=0; str[index]!='\0'; index++)
    {
        ascii[str[index]]++;
        if(ascii[str[index]] == 2)
        {
            printf("%c",str[index]);
            return;
        }

    }
    return 0;
}
```

Code:

```c
#include<stdio.h>
#include<stdlib.h>

int main()
{
    char str[100000];
    scanf("%s",str);
    int ascii[128]={0};
    for(int i=0;str[i]!='\0';i++){ //if null it will exist the loop
        if(ascii[str[i]] ==1){
            printf("%c",str[i]);
            return;
        }
        ascii[str[i]]++;
    }
}
```

Session 2:



Stimulation:

https://pythontutor.com/visualize.html#code=%23include%20%3Cstdio.h%3E%0A%0Avoid%20print%28int%20val%29%7B%0A%20%20%20%20if%28val%3D%3D0%29%7B%0A%20%20%20%20%20%20%20%20return%3B%0A%20%20%20%20%7D%0A%20%20%20%20print%28val-1%29%3B%0A%20%20%20%20printf%28%22%25d%20%22,val%29%3B%0A%7D%0A%0Aint%20main%28%29%7B%0A%20%20%20%20int%20N%3D5%3B%0A%20%20%20%20print%28N%29%3B%0A%20%20%20%20return%200%3B%0A%7D&cumulative=false&curInstr=33&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false

Code:

```c
#include <stdio.h>

void print(int val){
    if(val==0){
        return;
    }
    print(val-1);
    printf("%d ",val);
}

int main(){
    int N=5;
    print(N);
    return 0;
}
```

Reconnect

```c
#include<stdio.h>

void print(int val){
    if(val==0){
        return;
    }
    printf("%d ",val);
    print(val-1);
}
int main()
{
    int N;
    scanf("%d",&N);
    print(N);
    return 0;
}
```


Recursion - Print from N to 1

ID:11081 Solved By 952 Users

The program accepts an integer **N** as the input and prints the integers from N to 1 as the output. Fill in the blanks with code so that the program runs successfully.

Boundary Condition(s):
1 <= N <= 1000

Example Input/Output 1:
Input:
4

Output:
4 3 2 1

Explanation:
The integers from 4 to 1 are printed as the output.

Example Input/Output 2:
Input:
9

Output:
9 8 7 6 5 4 3 2 1

Max Execution Time Limit: 500 millisecs

Ambiance C (gcc 8.x)

```c
#include<stdio.h>

void print(int val)
{
    if( val==0 )
    {
        return ;
    }
    printf( "%d",val );
    print( val-1 );
}

int main()
{
    int N;
    scanf("%d",&N);
    print(N);
    return 0;
}
```

19it019@tce

Integer to Binary Representation

ID:11082 Solved By 947 Users

The program accepts an integer **N** and prints the binary representation of N as the output. Fill in the blanks with code so that the program runs successfully.

Example Input/Output 1:
Input:
9

Output:
1001

Example Input/Output 2:
Input:
26

Output:
11010

Max Execution Time Limit: 500 millisecs

Ambiance C (gcc 8.x)

```c
#include<stdio.h>

void toBinary(int val)
{
    if(val==0)
    {
        return;
    }
    toBinary(val/2);
    printf("%d",val%2);
}

int main()
{
    int N;
    scanf("%d",&N);
    toBinary(N);
    return 0;
}
```

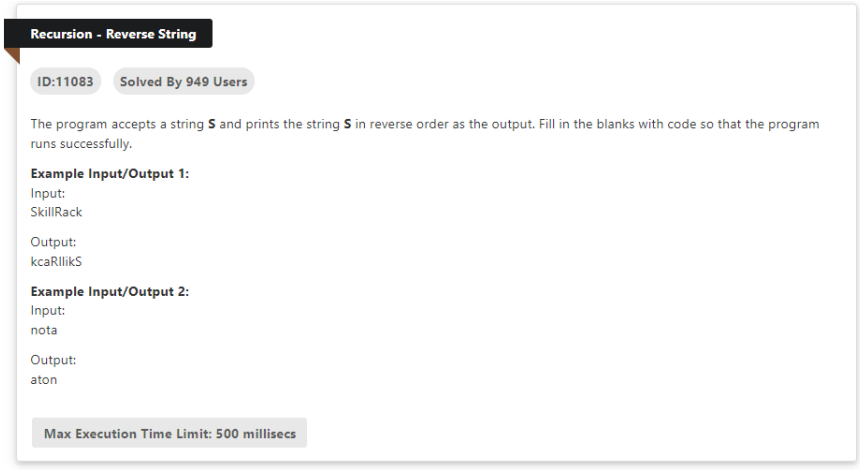
19it019@tce

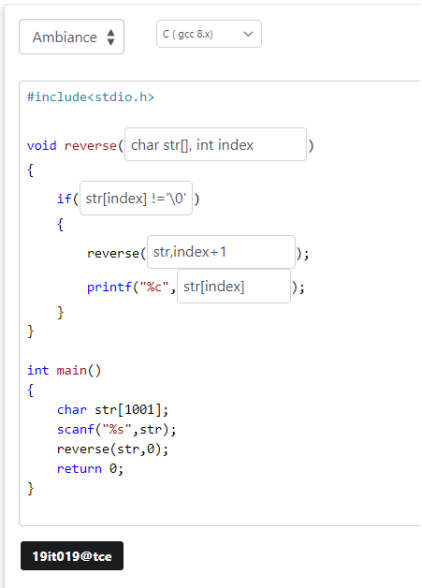
Stimulation:

https://pythontutor.com/visualize.html#code=%23include%20%3Cstdio.h%3E%0A%0Avoid%20print%28int%20val%29%7B%0A%20%20%20%20if%28val%3D%3D0%29%7B%0A%20%20%20%20%20%20%20%20return%3B%0A%20%20%20%20%7D%0A%20%20%20%20print%28val/2%29%3B%0A%20%20%20%20printf%28%22%25d%20%22,val%252%29%3B%0A%7D%0A%0Aint%20main%28%29%7B%0A%20%20%20%20int%20N%3D9%3B%0A%20%20%20%20print%28N%29%3B%0A%20%20%20%20return%200%3B%0A%7D&cumulative=false&curInstr=28&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false

Code:

```c
#include <stdio.h>

void print(int val){
    if(val==0){
        return;
    }
    print(val/2);
    printf("%d ",val%2);
}

int main(){
    int N=9;
    print(N);
    return 0;
}
```



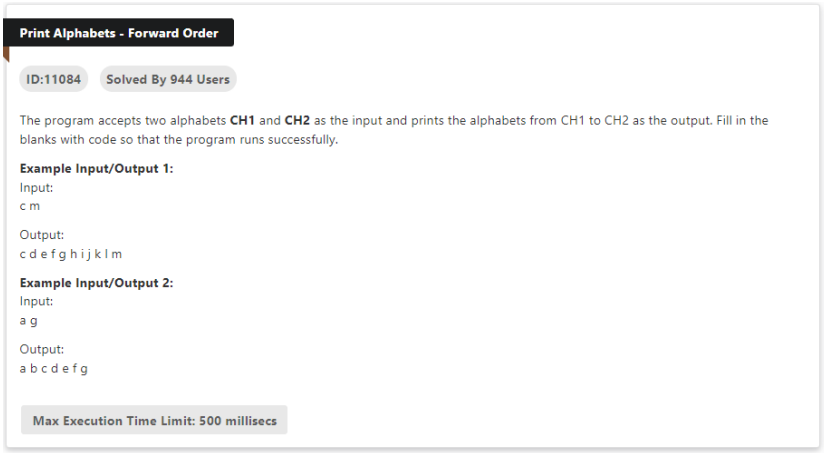


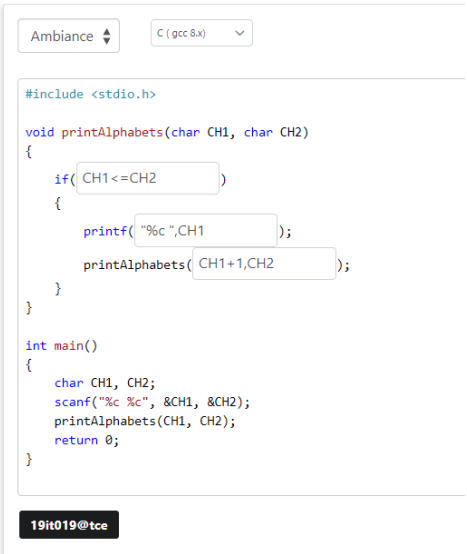
Stimulation:

https://pythontutor.com/visualize.html#code=%23include%20%3Cstdio.h%3E%0A%0Avoid%20reverse%28char%20str%5B%5D,int%20index%29%7B%0A%20%20%20%20if%28str%5Bindex%5D!%3D'%5C0'%29%7B%0A%20%20%20%20%20%20reverse%28str,index%2B1%29%3B%0A%20%20%20%20%20%20printf%28%22%25c%22,str%5Bindex%5D%29%3B%0A%20%20%20%20%7D%0A%7D%0A%0Aint%20main%28%29%7B%0A%20%20%20%20char%20str%5B%5D%3D%22Days%22%3B%0A%20%20%20%20reverse%28str,0%29%3B%0A%20%20%20%20return%200%3B%0A%7D&cumulative=false&curInstr=32&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false

Code:

```c
#include <stdio.h>

void reverse(char str[],int index){
    if(str[index]!='\0'){
      reverse(str,index+1);
      printf("%c",str[index]);
    }
}

int main(){
    char str[]="SkillRack";
    reverse(str,0);
    return 0;
}
```



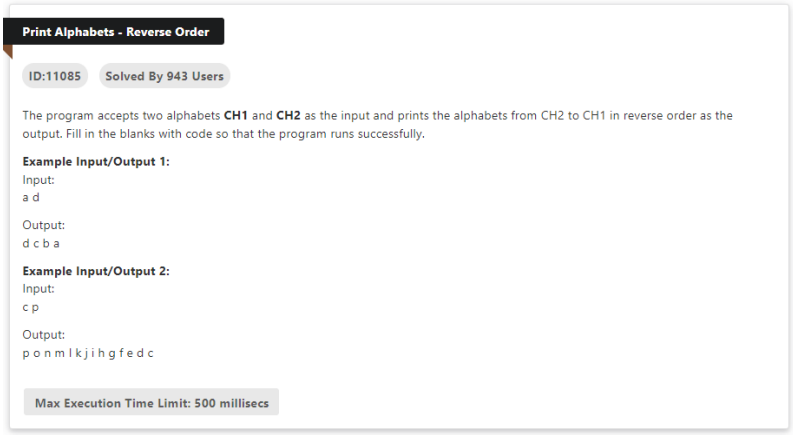


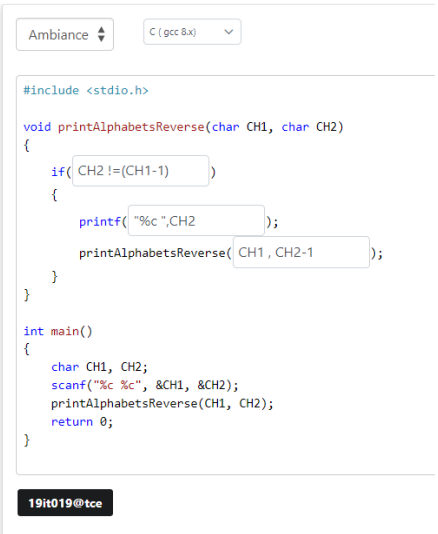
Stimulation:

https://pythontutor.com/visualize.html#code=%23include%20%3Cstdio.h%3E%0A%0Avoid%20printAlphabets%28char%20CH1,char%20CH2%29%7B%0A%20%20%20%20if%28CH1%3C%3DCH2%29%7B%0A%20%20%20%20%20%20printf%28%22%25c%20%22,CH1%29%3B%0A%20%20%20%20%20%20printAlphabets%28CH1%2B1,CH2%29%3B%0A%20%20%20%20%20%20%0A%20%20%20%20%7D%0A%7D%0A%0Aint%20main%28%29%7B%0A%20%20%20%20char%20CH1%3D'a'%3B%0A%20%20%20%20char%20CH2%3D'd'%3B%0A%20%20%20%20printAlphabets%28CH1,CH2%29%3B%0A%20%20%20%20return%200%3B%0A%7D&cumulative=false&curInstr=0&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false

Code:

```c
#include <stdio.h>

void printAlphabets(char CH1,char CH2){
    if(CH1<=CH2){
      printf("%c ",CH1);
      printAlphabets(CH1+1,CH2);

    }
}

int main(){
    char CH1='a';
    char CH2='g';
    printAlphabets(CH1,CH2);
    return 0;
}
```





Stimulation:

https://pythontutor.com/visualize.html#code=%23include%20%3Cstdio.h%3E%0A%0Avoid%20printAlphabetsReverse%28char%20CH1,char%20CH2%29%7B%0A%20%20%20%20if%28CH2!%3D%28CH1-1%29%29%7B%0A%20%20%20%20%20%20printf%28%22%25c%20%22,CH2%29%3B%0A%20%20%20%20%20%20printAlphabetsReverse%28CH1,CH2-1%29%3B%0A%20%20%20%20%7D%0A%7D%0A%0Aint%20main%28%29%7B%0A%20%20%20%20char%20CH1%3D'a'%3B%0A%20%20%20%20char%20CH2%3D'd'%3B%0A%20%20%20%20printAlphabetsReverse%28CH1,CH2%29%3B%0A%20%20%20%20return%200%3B%0A%7D&cumulative=false&curInstr=34&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false

Code:

```c
#include <stdio.h>

void printAlphabetsReverse(char CH1,char CH2){
    if(CH2!=(CH1-1)){
      printf("%c ",CH2);
      printAlphabetsReverse(CH1,CH2-1);
    }
}

int main(){
    char CH1='a';
    char CH2='g';
    printAlphabetsReverse(CH1,CH2);
    return 0;
}
```

TO ANY BASE:

```c
#include<stdio.h>
char digits[] = "0123456789ABCDEF";

void toBase(int val, int base){
    if(val==0){
```

```
        return;
    }
    toBase(val/base,base);
    printf("%c",digits[val%base]);
}

int main(){
    int N=14;
    int base=2;
    toBase(N,base);
    return 0;
}
```

Extras:

```
{
    int N;
    scanf("%i",&N);
    printf("%d", N);
    return 0;
}
```

Your Input

015

Your Program Output:

13

Save Run

```
#include<stdio.h>

int main()
{
    int N;
    scanf("%i",&N);
    printf("%d", N);
    return 0;
}
```

Your Input

0x10

Your Program Output:

16

Save Run

SESSION 3:

Permutations - String Characters

ID:11086 Solved By 910 Users

The program must accept a string **S** as the input. The program must print all the permutations of the string S as the output.

Boundary Condition(s):
2 <= Length of S <= 10

Input Format:
The first line contains S.

Output Format:
The lines containing all the permutations of the string S.

Example Input/Output 1:
Input:
abc

Output:
abc
acb
bac
bca
cba
cab

Example Input/Output 2:
Input:
rack

Output:
rack
rakc
rcak
rcka
rkca
rkac
arck
arkc
acrk
ackr
akcr
akrc
cark
cakr
crak
crka
ckra
ckar
kacr
karc
kcar
kcra
krca
krac

Max Execution Time Limit: 1000 millisecs

Code:

```
#include<stdio.h>
#include<stdlib.h>

void swap(char str[],int x,int y){
    char temp=str[x];
    str[x]=str[y];
    str[y]=temp;
}
void permute(char str[], int left,int right){
    if(left==right){
        printf("%s\n",str);
        return;
    }
```

```
    for(int index=left;index<=right;index++){
        swap(str,left,index);
        permute(str,left+1,right);
        swap(str,left,index);
    }}
int main()
{
    char str[50];
    int len;
    scanf("%s%n",str,&len);
    permute(str,0,len-1);
    return 0;
}
```

Problem 2:

Permutation Nearest Value

ID:11087 **Solved By 859 Users**

The program must accept two integers **N** and **X** as the input. The program must print the integer value nearest to X with all the digits in N as the output.

Boundary Condition(s):
10 <= N, X <= 10^8

Input Format:
The first lines contains N.

Output Format:
The first lines contains the integer value nearest to X with all the digits in N as the output.

Example Input/Output 1:
Input:
123 200

Output:
213

Explanation:
The integer value nearest to the **200** with all the digits in 123 is **213**.

Example Input/Output 2:
Input:
48871 88555

Output:
88471

Max Execution Time Limit: 1000 millisecs

Code:

```c
#include<stdio.h>
#include<stdlib.h>
int x,closest;
void swap(char str[],int x,int y){
    char temp=str[x];
    str[x]=str[y];
    str[y]=temp;
}

int absDiff(int a,int b){
    return a>b? a-b:b-a;
}

void permute(char str[],int left,int right){
    if(left==right){
        int curr = atoi(str);
        if(absDiff(x,curr) < absDiff(x,closest)){
            closest=curr;
        }
        return;
    }
    for(int index= left;index<=right;index++){
        swap(str,left,index);
        permute(str,left+1,right);
        swap(str,left,index);
    }
}

int main()
{
    char str[50];
    int len;
    scanf("%s%n%d",str,&len,&x);
    closest = atoi(str);
    permute(str,0,len-1);
    printf("%d",closest);
    return 0;
}
```