Elite-S029-Numbers

Solved Challenges **1/2**

Back To Challenges List

Smallest Possible Odd Integer

ID:11124 **Solved By 633 Users**

The program must accept an integer **N** as the input. The program must print the smallest possible odd integer using all the digits in N as the output. If it is not possible to form such an integer, the program must print no as the output.

Boundary Condition(s):
10 <= N <= 10^17

Input Format:
The first line contains N.

Output Format:
The first line contains the smallest possible odd integer using all the digits in N or no.

Example Input/Output 1:
Input:
1670078423

Output:
1002346787

Explanation:
The smallest possible odd integer using all the digits in 1670078423 is **1002346787**.

Example Input/Output 2:
Input:
40068

Output:
no

Max Execution Time Limit: 500 millisecs

Ambiance

C (gcc 8.x)

Reset

```c
#include<stdio.h>
#include<stdlib.h>
#define ULL unsigned long long int

int main()
{
    ULL N;
    scanf("%llu",&N);
    int digits[10]={0};
    while(N!=0)
    {
        digits[N%10]++;
        N/=10;
    }
    int unitDig=-1;
    for(int dig=9;dig>=1;dig-=2)
    {
        if(digits[dig]>0)
        {
            unitDig=dig;
            digits[dig]--;
            break;
        }
    }
    if(unitDig==-1)
    {
        printf("no");
        return;
    }
    int start = 1;
    for(int digit=1;digit<=9;digit++)
    {
        if(digits[digit]>0)
        {
        printf("%d",digit);
        digits[digit]--;
        start=0;
        break;
        }
    }

    for(int digit=start;digit<=9;digit++)
    {
        while(digits[digit]-->0)
            printf("%d",digit);
    }

    printf("%d",unitDig);
}
```

SkillRack Message : Smallest Possible Odd Integer

Close

```
#include<stdio.h>
#include<stdlib.h>
#define ULL unsigned long long int

int main()
{
ULL N;
scanf("%lld",&N);
int digits[10]={0};
while(N!=0)
{
digits[N%10]++;
N/=10;
}
int unitDig=-1;
for(int dig=9;dig>=1;dig-=2)
{
if(digits[dig]>0)
{
unitDig=dig;
digits[dig]--;
break;
}
}
if(unitDig==-1)
{
printf("no");
return;
}
int start = 1;
for(int digit=1;digit<=9;digit++)
{
if(digits[digit]>0)
{
printf("%d",digit);
digits[digit]--;
start=0;
break;
}
}

for(int digit=start;digit<=9;digit++)
{
while(digits[digit]-->0)
printf("%d",digit);
}

printf("%d",unitDig);
}
```

SkillRack Message : Smallest Possible Odd Integer

```
#include<stdio.h>
#include<stdlib.h>
#define ULL unsigned long long int

int main()
```

```
{
ULL N;
scanf("%lld",&N);
int digits[10]={0};
while(N!=0)
{
digits[N%10]++;
N/=10;
}
int unitDig=-1;
for(int dig=9;dig>=1;dig-=2)
{
if(digits[dig]>0)
{
unitDig=dig;
digits[dig]--;
break;
}
}
if(unitDig==-1)
{
printf("no");
return;
}
int start = 1;
for(int digit=1;digit<=9;digit++)
{
if(digits[digit]>0)
{
printf("%d",digit);
digits[digit]--;
start=0;
break;
}
}

for(int digit=start;digit<=9;digit--)
{
while(digits[digit]-->0)
printf("%d",digit);
}

printf("%d",unitDig);
}
```

Code did not pass the execution

TestCase ID: 63903

Input:

1670078423

Expected Output:

1002346787

Your Program Output:

Your program did not execute successfully:
SoftTimeLimitExceeded()

Save Run

Run with a custom test case (Input/Output)