

GE23131-Programming Using C-2024

Status

Finished

Started

Monday, 23 December 2024, 5:33 PM

Completed

Tuesday, 3 December 2024, 1:56 PM


Duration

20 days 3 hours

Question 1

Correct

Marked out of 3.00

 Flag question

Question text

The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.

Given a positive integer N, return true if and only if it is an Armstrong number.

Example 1:

Input:

153

Output:

true

Explanation:

153 is a 3-digit number, and $153 = 1^3 + 5^3 + 3^3$.

Example 2:

Input:

123

Output:

false

Explanation:

123 is a 3-digit number, and $123 != 1^3 + 2^3 + 3^3 = 36$.

Example 3:

Input:

1634

Output:

true

Note:

$1 \leq N \leq 10^8$

Answer:(penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include<math.h>
3
4 int main(){
5     int n,a,p=0,t,sum=0,x;
6     scanf("%d",&n);
7     t=n;
8     a=n;
9     while(t!=0){
10         t=t/10;
11         p++;
12     }
13     while(n!=0){
14         x=n%10;
15         sum=sum+pow(x,p);
16         n=n/10;
17     }
18     if(sum==a){
19         printf("true");
20     }else{
21         printf("false");
22     }
23     return 0;
24
25
26
27 }
```

Feedback

Input Expected Got


153	true	true
123	false	false

Passed all tests!

Question 2

Correct

Marked out of 5.00

 Flag question

Question text

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints $1 \leq \text{num} \leq 99999999$ Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066
Answer:(penalty regime: 0 %)

```
1 #include <stdio.h>
2 int main(){
3     long long int num,sum,rev,tempnum,tempsum;
4     scanf("%lld",&num);
5     while(1){
6         rev=0;
7         tempnum=num;
8         while(num){
9             rev=rev*10+(num%10);
10            num=num/10;
11        }
12        sum=tempnum+rev;
13        tempsum=sum;
14        rev=0;
15        while(sum){
16            rev=rev*10+(sum%10);
17            sum=sum/10;
18        }if(tempsum==rev)
19            break;
20        num=tempsum;
21    }
22
23    printf("%lld",tempsum);
24    return 0;
25 }
26 }
```

Feedback


Input Expected Got

32	55	55
789	66066	66066

Passed all tests!

Question 3

Correct
Marked out of 7.00

 Flag question

Question text

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample Input 1:

3

Sample Output 1:

33

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2:

33344

Answer:(penalty regime: 0 %)

```
1 #include <stdio.h>
2 int main(){
3     long int i,j;
4     int rem,n,count=0;
5     int flag;
6     scanf("%d",&n);
7     for(i=1;count<=n;i++){
8         flag=0;
9         j=i;
10        while(j>0){
11            rem=j%10;
12            if(rem==3 || rem==4)
13                j=j/10;
14        }
15        else{
16            flag=1;
17            break;
18        }
19        if(flag==0){
20            count++;
21            if(count==n)
22                break;
23        }
24    }
25    printf("%ld",i);
26    return 0;
27 }
```

Feedback

Input Expected Got

34	33344	33344
----	-------	-------

Passed all tests!

Finish review

Blocks

[Skip Quiz navigation](#)

Quiz navigation

[Question 1 This page](#) [Question 2 This page](#) [Question 3 This page](#)

[Show one page at a time](#)

Finish review

Blocks