GE23131-Programming Using C-2024

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Sunday, 22 December 2024, 12:28 PM
Duration	1 day 5 hours

Question **1**

Correct

Marked out of 3.00

Flag question

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 <= N <= 10^8

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
int main()
{
   int num,o_num,rem,r=0,n=0;
   scanf("%d",&num);
   o_num=num;

   while (o_num !=0) {
       o_num /= 10;
       n++;
   }
    o_num = num;

   while (o_num != 0) {
       rem = o_num % 10;
       r += pow(rem,n);
       o_num /= 10;
   }

   if (r == num)
   printf("true\n");
   else
   printf("false\n");

   return 0;
}
```

	Input	Expected	Got	
✓	153	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints 1<=num<=99999999 Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
long long reverse ( long long num)
{
long long reversed = 0;
while (num != 0)
{
reversed = reversed * 10 + num % 10;
num /= 10;
}
return reversed;
}

int isPalindrome ( long long num)
{
return num == reverse(num);
}
int main()
{
long long num, reversed,sum;
scanf("%lld", &num);
if (num < 1 || num > 999999999)
{
printf("no. out range");
return 1;
}
while (1)
{
reversed = reverse(num);
sum = num + reversed;

if (isPalindrome(sum))
{
printf("%lld\n",sum);
break;
}
num = sum;
}
return 0;
}
```

	Input	Expected	Got	
✓	32	55	55	✓
✓	789	66066	66066	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 7.00

Flag question

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 %)

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

bool isLucky(int n) {
    while (n > 0) {
        int digit = n % 10;
        if (digit != 3 && digit != 4) {
            return false;
        }
        n /= 10;
    }
    return true;
}
int findNthLucky(int n) {
    int count = 0,num = 1;
    while (count < n) {
        if (isLucky(num)) {
            count ++;
        }
        num++;
    }
    return num - 1;
}

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

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓

Finish review