# REC-CIS Maanisha E - 240901055

GE23131-Programming Using C-2024

|  |  |
| --- | --- |
| **Status** | Finished |
| **Started** | Monday, 23 December 2024, 5:33 PM |
| **Completed** | Saturday, 7 December 2024, 1:05 PM |
| **Duration** | 16 days 4 hours |

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



Question **1**

Correct

Marked out of 3.00

Flag question

Quiz navigation

1

2

3

[Show one page at a time](http://www.rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=111561&cmid=123&showall=0)

Finish review

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

Example 1:

Input:

153

Output:

true

Explanation:

REC-CIS

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

REC-CIS

Note:

1 <= N <= 10^8

**Answer:** (penalty regime: 0 %)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 |  |  | #include<stdio.h> |  |
| 2 |  |  | #include<math.h> |  |
| 3 |  |  | int main() |  |
| 4 | ▼ |  | { |  |
| 5 |  |  | int n; |  |
| 6 |  |  | scanf("%d",&n); |  |
| 7 |  |  | int x=0,n2=n; |  |
| 8 |  |  | while(n2!=0) |  |
| 9 | ▼ |  | { |  |
| 10 |  |  | x++; |  |
| 11 |  |  | n2=n2/10; |  |
| 12 |  |  | } |  |
| 13 |  |  | int sum=0; |  |
| 14 |  |  | int n3=n,n4; |  |
| 15 |  |  | while(n3!=0) |  |
| 16 | ▼ |  | { |  |
| 17 |  |  | n4=n3%10; |  |
| 18 |  |  | sum=sum+pow(n4,x); |  |
| 19 |  |  | n3=n3/10; |  |
| 20 |  |  | } |  |
| 21 |  |  | if(n==sum) |  |
| 22 | ▼ |  | { |  |
| 23 |  |  | printf("true"); |  |
| 24 |  |  | } |  |
| 25 |  |  | else |  |
| 26 | ▼ |  | { |  |
| 27 |  |  | printf("false"); |  |
| 28 |  |  | } |  |
| 29 |  |  | return 0; |  |
| 30 |  |  | } |  |
|  |  |  |  |  |

# REC-CIS

**Input Expected Got**



|  |  |  |
| --- | --- | --- |
| 153 | true | true |
| 123 | false | false |

Passed all tests!

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 %)



Question **2**

Correct

Marked out of 5.00

Flag question

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 |  |  | #include<stdio.h> |  |
| 2 |  |  | int main() |  |
| 3 | ▼ |  | { |  |
| 4 |  |  | int rn,n,nt=0,i=0; |  |
| 5 |  |  | scanf("%d",&n); |  |
| 6 |  |  | do |  |
| 7 | ▼ |  | { |  |
| 8 |  |  | nt=n;rn=0; |  |
| 9 |  |  | while(n!=0) |  |
| 10 | ▼ |  | { |  |
| 11 |  |  | rn=rn\*10+n%10; |  |
| 12 |  |  | n=n/10; |  |
| 13 |  |  | } |  |
| 14 |  |  | n=nt+rn; |  |
| 15 |  |  | i++; |  |
| 16 |  |  | } |  |
| 17 |  |  | while(rn!=nt||i==1); |  |
| 18 |  |  | printf("%d",rn); |  |
| 19 |  |  | return 0; |  |
| 20 |  |  | } |  |
|  |  |  |  |  |

# REC-CIS

**Input Expected Got**



|  |  |  |
| --- | --- | --- |
| 32 | 55 | 55 |
| 789 | 66066 | 66066 |

Passed all tests!

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.



Question **3**

Correct

Marked out of 7.00

Flag question

# REC-CIS

Sample Input 2:

34

Sample Output 2:

33344

**Answer:** (penalty regime: 0 %)

1

#include<stdio.h>

2

3 ▼

4

5

6

7 ▼

8

9

10 ▼

11

12

13 ▼

14

15

16

17

18

19

20 ▼

21

int main()

{

int n=1,i=0,nt,co=0,e; scanf("%d",&e);

while(i<e)

{

nt=n;

while(nt!=0)

{

co=0;

if(nt%10!=3 && nt%10!=4)

{

co=1; break;

}

nt=nt/10;

}

if(co==0)

{

i++;

# REC-CIS

22 }

23 n++;

24 }



1. printf("%d",--n);
2. return 0;
3. }

28

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 34 | 33344 | 33344 |  |

Passed all tests!

Finish review