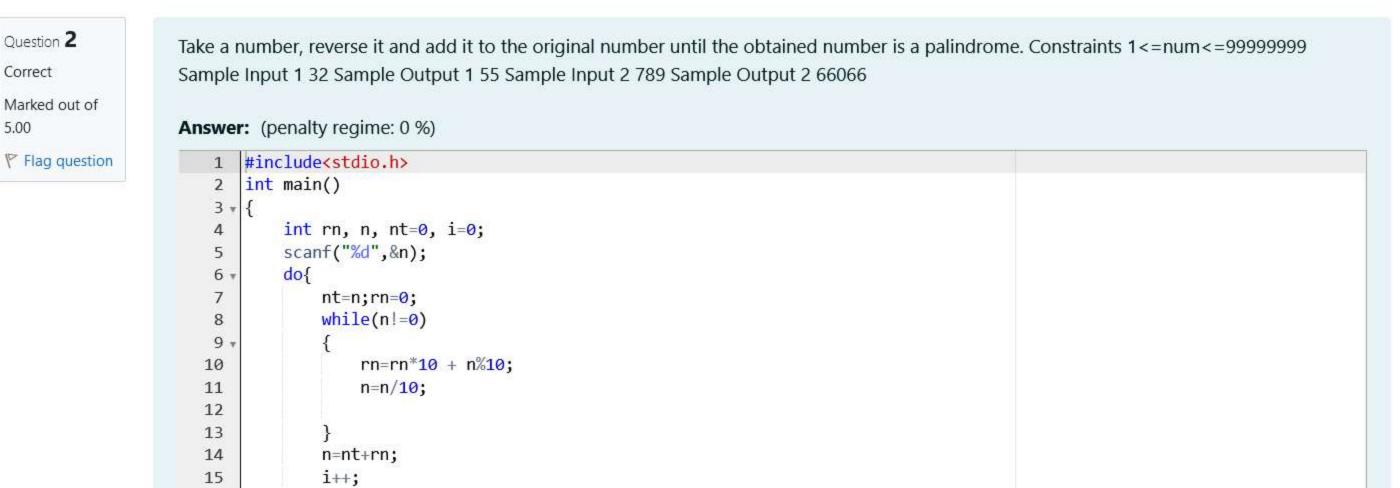
Question 1 The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N. Correct Marked out of 3.00 Given a positive integer N, return true if and only if it is an Armstrong number. Flag question Example 1: Input: 153 Output: true Explanation: 153 is a 3-digit number, 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 %) 1 #include<stdio.h> #include<math.h> int main() 3 4 + { int n; 5 scanf("%d",&n); 6 int x=0,n2=n; 7 while(n2!=0) 8 9 * 10 X++;n2=n2/10;11 12 13 int sum=0; 14 int n3=n,n4; 15 while(n3!=0) 16 17 * 18 n4=n3%10; sum=sum+pow(n4,x); 19 n3=n3/10; 20 21 22 if(n==sum) 23 24 1 printf("true"); 25 26 27 else 28 + printf("false"); 29 30



Question 2

Correct

5.00

return 0;

Input Expected Got

true

false

true

false

31 32 }

153

123

Passed all tests! <

16

17

```
while(rn!=nt || i==1);
         printf("%d",rn);
 18
         return 0;
 19
 20
     Input Expected Got
            55
                      55
                             ~
     32
     789
            66066
                      66066
                             ~
Passed all tests! <
```

Question 3

Marked out of

Flag question

Correct

7.00

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:

33 Explanation:

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

34

3

Sample Output 1:

Sample Input 2:

Answer: (penalty regime: 0 %)

1 #include<stdio.h>

int n=1,i=0,nt,co=0,e;

while(nt!=0)

co=0;

co=1;

break;

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

scanf("%d",&e);

nt=n;

while(i<e)

int main()

2

4

5

9 10 11

12 13 1

14 15

16 17

3 + {

Sample Output 2: 33344

```
nt=nt/10;
 18
 19
             if(co==0)
 20
 21 *
                 i++;
 22
 23
             n++;
 24
 25
         printf("%d",--n);
 26
         return 0;
 27
 28 }
     Input Expected Got
     34
            33344
                      33344
Passed all tests! <
```