Week-05-02-Practice session-Coding

| Question 1 Correct | The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N. |
|---------------------------------------|--|
| Marked out of 3.00 ♥ Flag question | 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: |

```
Output:

true

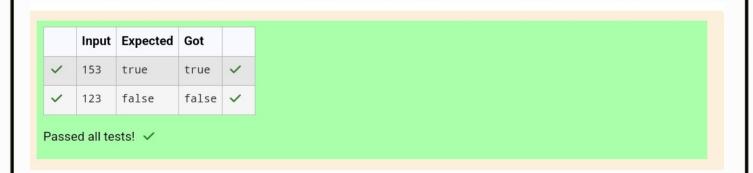
Note:

1 <= N <= 10^8
```

Source code

```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
   2 v int main(){
   3
           int n,k=0,a=0;
           scanf("%d",&n);
   4
   5
           int n1=n,n2=n;
           while(n1>0){
   6 *
               k+=1;
   7
   8
               n1/=10;
   9
           while(n2>0){
  10 *
               int b=n2%10,c=1;
  11
               for(int i=1;i<=k;i++){
  12 🔻
  13
                   c*=b;
  14
               }
  15
               a+=c;
  16
               n2=n2/10;
  17
           if (a==n){
  18 *
               printf("true");
  19
  20
           else{
  21 🔻
               printf("false");
  22
  23
  24
           return 0;
  25 }
```

Result



Question **2**Correct
Marked out of 5.00

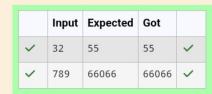
Frag question

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 %)

```
1 |#include<stdio.h>
    int main(){
        long long int n,s,rev,temp1,temp2;
3
 4
        scanf("%lld",&n);
5 ,
        while(1){
 6
            temp1=n,rev=0;
            while(n){
                rev=rev*10+(n%10);
8
 9
                n=n/10;
10
            s=temp1+rev;
11
            temp2=s,rev=0;
12
13
            while(s){
                rev=rev*10+(s%10);
14
                s=s/10;
15
16
17 -
            if(temp2==rev){
18
                break;
19
20
            n=temp2;
21
        printf("%lld",temp2);
22
23
        return 0;
24 }
```

Result



Passed all tests! 🗸

Question **3**Correct
Marked out of 7.00
Friag 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

Source code

```
#include<stdio.h>
    int main(){
 3
         int n;
         scanf("%d",&n);
 4
 5
         int x[n], c=0, g=0, s=0;
 6 *
        while(c<n){</pre>
 7
             int r;
             (c%2==0)?(r=3):(r=4);
 8
 9 ,
             if ((c\%2==0)\&\&(c!=0)){
10
                 s=x[g];
11
                 g++;
12
13
             x[c]=(s*10)+r;
14
             C++;
15
         printf("%d",x[n-1]);
16
17
        return 0;
18 }
```

Result

