

Status	Finished
Started	Sunday, 2 November 2025, 8:05 PM
Completed	Sunday, 2 November 2025, 10:00 PM
Duration	1 hour 55 mins

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.

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 $123 \neq 1^3 + 2^3 + 3^3 = 36$.

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  int main()
4
5  {
6  long long int num,sum=0,nod=0,rem,temp;
7  scanf("%lld",&num);
8  temp=num;
9  while(num>0)
10 {
11     nod++;
12     num=num/10;
13 }
14 num=temp;
15 while(num>0)
16 {
17     rem=num%10;
18     sum=sum+pow(rem,nod);
19     num=num/10;
20 }
21 if(sum==temp)
22     printf("true");
23 else
24     printf("false");
25 return 0;
26 }
```



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

Passed all tests! ✓

Question **2**

Correct

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

For example:

Input	Result
32	55
1234	5555

Answer: (penalty regime: 0 %)

```
1  #include<stdio.h>
2  int reverse(int n)
3  {
4      int rev=0;
5      while(n>0)
6      {
7          rev=rev*10+n%10;
8          n=n/10;
9      }
10     return rev;
11 }
12
13 int main()
14 {
15     int num;
16     scanf("%d",&num);
17     while(num!=reverse(num))
18     {
19         num=num+reverse(num);
20     }
21     printf("%d",num);
22     return 0;
23 }
```



	Input	Expected	Got	
✓	32	55	55	✓
✓	1234	5555	5555	✓

Passed all tests! ✓



Question **3**

Correct

Maya, a student in an arts and crafts class, wants to create a pattern using stars (*) in a specific format. She plans to use a program to help her construct the pattern.

Write a program that takes an integer as input and constructs the following pattern using nested for loops.

Input: 5

Output:

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Answer: (penalty regime: 0 %)

```
1  #include<stdio.h>
2  int main(void)
3  {
4      int n,i,j;
5      scanf("%d",&n);
6      for (i=1; i<=n; i++){
7          for(j=1;j<=i;j++){
8
9              printf("*");
10             if(j<i)printf(" ");
11         }
12         printf("\n");
13     }
14     for(i=n-1;i>=1;i--){
15         for (j=1;j<=i;j++){
16             printf("*");
17             if(j<i)printf(" ");
18         }
19         if(i>1)printf("\n");
20     }
21     return 0;
22 }
```



	Input	Expected	Got	
✓	5	<pre>* *</pre>	<pre>* *</pre>	✓

Passed all tests! ✓