

<b>Status</b>	Finished
<b>Started</b>	Friday, 31 October 2025, 6:14 PM
<b>Completed</b>	Friday, 31 October 2025, 7:18 PM
<b>Duration</b>	1 hour 3 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  $1^3 + 5^3 + 3^3 = 153$ .

Example 2:

Input:

123

Output:

false

Explanation:

123 is a 3-digit number, and  $1^3 + 2^3 + 3^3 \neq 123$ .

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     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 }
31

```

	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 main()
3 {
4     int rn,n,nt=0,i=0;
5     scanf("%d",&n);
6     do{
7         nt=n;rn=0;
8         while(n!=0)
9         {
10            rn=rn*10+n%10;
11            n=n/10;
12        }
13        n=nt+rn;
14        i++;
15    }
16    while(rn!=nt||i==1);
17    printf("%d",rn);
18    return 0;
19 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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(){
3     int n,i,j;
4     scanf("%d",&n);
5     for(i=1;i<=n;i++){
6         for(j=1;j<=i; j++){
7             printf("* ");
8         }
9         printf("\n");
10    }
11    for(i=n-1;i>=1;i--){
12        for(j=1;j<=i;j++){
13            printf("* ");
14        }
15        printf("\n");
16    }
17    return 0;
18
19
20 }
```



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

Passed all tests! ✓