

Status	Finished
Started	Monday, 3 November 2025, 5:12 PM
Completed	Monday, 3 November 2025, 5:37 PM
Duration	25 mins 3 secs

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     int n,i,nc,nc1,ct=0;
6     int rev=0;
7     scanf("%d",&n);
8     nc=n;
9     while(nc>0)
10    {
11         nc=nc/10;
12         ct++;
13     }
14     nc1=n;
15     while(nc1>0)
16    {
17         i=nc1%10;
18         rev = rev + (int)pow(i,ct);
19         nc1=nc1/10;
20     }
21     if (rev==n)
22    {
23         printf("true");
24     }
25     else
26    {
27         printf("false");
28     }
29     return 0;
30 }
```

	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 n,m;
5     int o=0;
6     int p=0;
7     scanf("%d",&n);
8     do
9     {
10         o=n;
11         m=0;
12         while(n!=0)
13         {
14             m=(m*10)+(n%10);
15             n=n/10;
16         }
17         n=o+m;
18         p++;
19     }
20     while(m!=o || p==1);

```

```
21     printf("%u", m);
22 }
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 ()
3 {
4     int n,i,j;
5     scanf("%d",&n);
6     for(i=1;i<=n;i++)
7     {
8         for(j=1;j<=i;j++)
9         {
10             printf("* ");
11         }
12         printf("\n");
13     }
14     for(i=n;i>=2;i--)
15     {
16         for(j=i;j>=2;j--)
17         {
18             printf("* ");
19         }
20         printf("\n");
21     }
22     return 0;
23 }
```

	Input	Expected	Got	
✓	5	*	*	✓
		* *	* *	
		* * *	* * *	
		* * * *	* * * *	
		* * * * *	* * * * *	
		* * * *	* * * *	
		* * *	* * *	
		* *	*	
		*		

Passed all tests! ✓