

Status	Finished
Started	Monday, 3 November 2025, 4:59 PM
Completed	Monday, 3 November 2025, 5:21 PM
Duration	21 mins 43 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 $1^3 + 5^3 + 3^3 = 153$.

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     int n,t,d=0,s=0;
5     scanf("%d",&n);
6     t=n;
7     while(t) {
8         d++; t/=10;
9         t=n;
10        while(t) {
11            int r=t%10;
12            s+=pow(r,d);
13            t/=10;
14        }
15        printf("%s", (s==n)? "true": "false");
16        return 0;
17
18
19 }
```



	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 rev(int n) {
3     int r=0;
4     while(n) {r=r*10+n%10;n/=10;}
5     return r;
6 }
7 int pal(int n) { return n==rev(n);}
8 int main() {
9     int n;
10    scanf("%d",&n);
11    while(!pal(n))n+=rev(n);
12    printf("%d",n);
13    return 0;
14 }
```

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

	Input	Expected	Got	
✓	5	* *	* *	✓

Passed all tests! ✓