

| | |
|------------------|-----------------------------------|
| Status | Finished |
| Started | Sunday, 2 November 2025, 10:45 PM |
| Completed | Sunday, 2 November 2025, 11:04 PM |
| Duration | 19 mins 4 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  int main()
3  {
4      int temp,n,org,r,result=0,digits=0;
5      scanf("%d",&n);
6      org=n;
7      temp=n;
8      while(temp!=0)
9      {
10         temp/=10;
11         digits++;
12     }
13     temp=n;
14     while(temp!=0)
15     {
16         r=temp%10;
17         int power=1;
18         for(int i=0;i<digits;i++)
19         {
20             power*=r;
21         }
22         result+=power;
23         temp/=10;
24     }
25     if(result==org)
26         printf("true");
27     else
28         printf("false");
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 $1 \leq \text{num} \leq 999999999$ **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 num,temp,rev,r;
5      scanf("%d",&num);
6      while(1)
7      {
8          temp=num;
9          rev=0;
10         while(temp!=0)
11         {
12             r=temp%10;
13             rev=rev*10+r;
14             temp/=10;
15         }
16         if(num==rev)
17             break;
18         num=num+rev;
19     }
20     printf("%d",num);
21     return 0;
22 }
```



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



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

Passed all tests! ✓