Status	Finished
Started	Tuesday, 4 November 2025, 9:28 AM
Completed	Tuesday, 4 November 2025, 9:42 AM
Duration	14 mins 26 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 != 1^3 + 2^3 + 3^3 = 36.

Example 3:

Input:

1634

Output:

true

Note:

1 <= N <= 10^8

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>

int main() {
    int n, i, nc, nc1, ct=0, rev=0;
    scanf("%d", &n);

    nc = n;

    while(nc>0) {
        nc=nc/10;
        ct++;
    }
    nc1=n;
    while(nc1>0) {
        i=nc1%10;
        rev = rev+ (int)pow(i, ct);
        nc1=nc1/10;
    }

    if (rev==n){
        printf("true");
    }else {
        printf("false");
    }

    return 0;
}
```

	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<=num<=99999999

Sample Input 1

32

Sample Output 1

55

For example:

Input	Result
32	55
1234	5555

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>

int main() {
    int n, m;
    int o=0;
    int p=0;
    scanf("%d", &n);

    do {
        o=n;
        m=0;
        while (n!=0) {
            m=(m*10)+(n%10);
            n = n/10;
        }
    n=o+m;
    p++;
    }while (m!=o || p==1);
    printf("%d", m);

    return 0;
}
```

	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 %)

```c
#include<stdio.h>

int main() {
    int n, i, j;
    scanf("%d", &n);

    for (i=1;i<=n;i++) {
        for (j=1; j<=i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    for (i=n; i>=2; i--) {
        for (j=i; j>=2; j--) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

	Input	Expected	Got	
✓	5	* *	* *	✓

Passed all tests! ✓