Week-05-02

Name: Mukesh V **Roll No: 241901059**

Armstrong Number

Problem Statement:

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.

Note: 1 <= N <= 10^8

Hint: 153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.

Sample Input:
153

Sample Output:
true

Sample **Input**:
123

Sample **Output**:
false

Sample **Input**:
1634

Sample **Output**:
true

Program:

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

int isArmstrong(int num){
    int originalNum = num;
    int sum = 0,digit,numDigits=0;

    int temp = num;
    while(temp!=0){
     numDigits++;
     temp /= 10;
    }
    temp = num;
    while(temp != 0){
        digit = temp % 10;
        sum += pow(digit,numDigits);
        temp /= 10;
    }
    return sum == originalNum;
}

int main(){
    int number;
    scanf("%d",&number);

    if(isArmstrong(number))
    printf("true");
    else
    printf("false");

    return 0;
}
```

	Input	Expected	Got	
✓	153	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

Reverse and Add Until Get a Palindrome

Problem Statement:

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

Sample Input 2
789

Sample Output 2
66066

Program:

```c
#include <stdio.h>

long long revNum(long long num){
    long long rev = 0;
    while(num>0){
        rev=rev*10+(num%10);
        num /= 10;
    }
    return rev;
}

int main(){
    long long num;
    scanf("%lld",&num);
    do{
        num += revNum(num);
    }
    while(num != revNum(num));
    printf("%lld\n",num);
    return 0;
}
```

	Input	Expected	Got	
✓	32	55	55	✓
✓	789	66066	66066	✓

Passed all tests! ✓

Lucky Number

Problem Statement:

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample Input 1:
3

Sample Output 1:
33

Program:

```c
#include <stdio.h>
int main(){
    int n=0,count=0,e,temp;
    scanf("%d",&e);
    while(count < e){
        n++;
        temp = n;
        while(temp > 0 && (temp%10==3 || temp % 10 == 4)){
            temp /=10;

        }
        if (temp == 0) count++;
    }
    printf("%d",n);
    return 0;
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓