

Week 5-2:

--Practice Session- Coding

ROLL NO.:240801178

Name: Logesh M

| | |
|------------------|-------------------------------------|
| Status | Finished |
| Started | Monday, 23 December 2024, 5:33 PM |
| Completed | Saturday, 14 December 2024, 2:04 PM |
| Duration | 9 days 3 hours |

Q1) 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: $1 \leq N \leq 10^8$

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

Code:

```

1 #include<stdio.h>
2 #include<math.h>
3 int main(){
4     int n,os,sum=0,digits=0;
5     scanf("%d",&n);
6     os=n;
7     int temp=n;
8     while(temp!=0){
9         temp/=10;
10        digits++;
11    }
12    temp=n;
13    while(temp!=0){
14        int r=temp%10;
15        sum +=pow(r,digits);
16        temp/=10;
17    }
18    if(sum==os)
19        printf("true");
20    else
21        printf("false");
22    }
23
24

```

Output:

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

Passed all tests! ✓

Q2) 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

Sample Input 2

789

Sample Output 2

66066

Code:

```
1 #include<stdio.h>
2 int reverse(int num){
3     int reversed=0;
4     while(num>0){
5         reversed=reversed*10+num%10;
6         num/=10;
7     }
8     return reversed;
9 }
10 int main(){
11     int num;
12     scanf("%d",&num);
13     do{
14         int rev=reverse(num);
15         num+=rev;
16     }while(num!=reverse(num));
17     printf("%d",num);
18 }
19
20
21
```

Output:

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

Passed all tests! ✓

Q3) 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:

Code:

```
1 #include<stdio.h>
2
3 int islucky(int n){
4     while (n>0){
5         int digit=n%10;
6         if(digit!=3&&digit!=4){
7             return 0;
8         }
9         n/=10;
10    }
11    return 1;
12 }
13 int main(){
14     int n;
15     scanf("%d",&n);
16     int count=0,number=1;
17     while(1){
18         if(islucky(number)){
19             count++;
20             if(count==n)
21                 break;
22         }
23         number++;
24     }
25     printf("%d",number);
26 }
27
28
29
```

Output:

| | Input | Expected | Got | |
|---|-------|----------|-------|---|
| ✓ | 34 | 33344 | 33344 | ✓ |

Passed all tests! ✓