Dhaka International University

Knowledge is Power
DIU
Estd. 7th April, 1995

DEPARTMENT OF CSE

LAB REPORT

COURSE NAME : Structured Programming Language Lab

COURSE CODE : 0613 - 102

REPORT NO : 08

REPORT ON : Reverse an Integer, Palindrome Number Armstrong number and Digit count in C Programming.

SUBMITTED BY	SUBMITTED TO
NAME : Mir Yeasir Abrar **ROLL** : 35 **REG. NO** : CS-D-98-23 - 127358 **BATCH** : 98 (1st shift) **SEMESTER** : 1st	Rakib Mahmud Lecturer Dept. of CSE

DATE OF SUBMISSION : 16 November, 2024

Date of Performance : 09 November, 2024

Title: Reverse an Integer, Palindrome Number, Armstrong Number and Digit Count in C Programming.

Objective: The objective of this lab report is to:

01. Implement a program to reverse the digit of an integer using arithmetic operations and loops.

02. Develop a program to determine if a given number is a palindrome by comparing the original number with its reversed version.

03. Write a program to check if a number is an Armstrong number by summing the power of its digits based on the total number of digits.

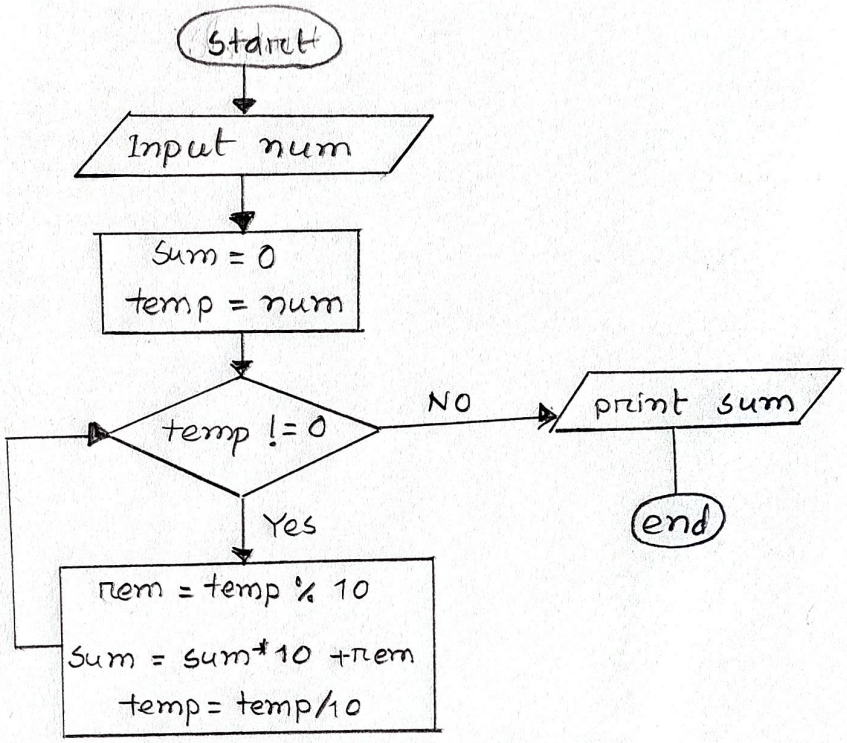
04. Creat a program to count the total number of digits in an integer through iterative division.

Introduction: This lab focuses on basic number manipulations using loops, conditional statements, and arithmetic operations in C Programming.

01. Reverse an Integer

... Explanation: The program extracts each digit of the number using modulus (%) and appends it to a reversed number.

... Flowchart:



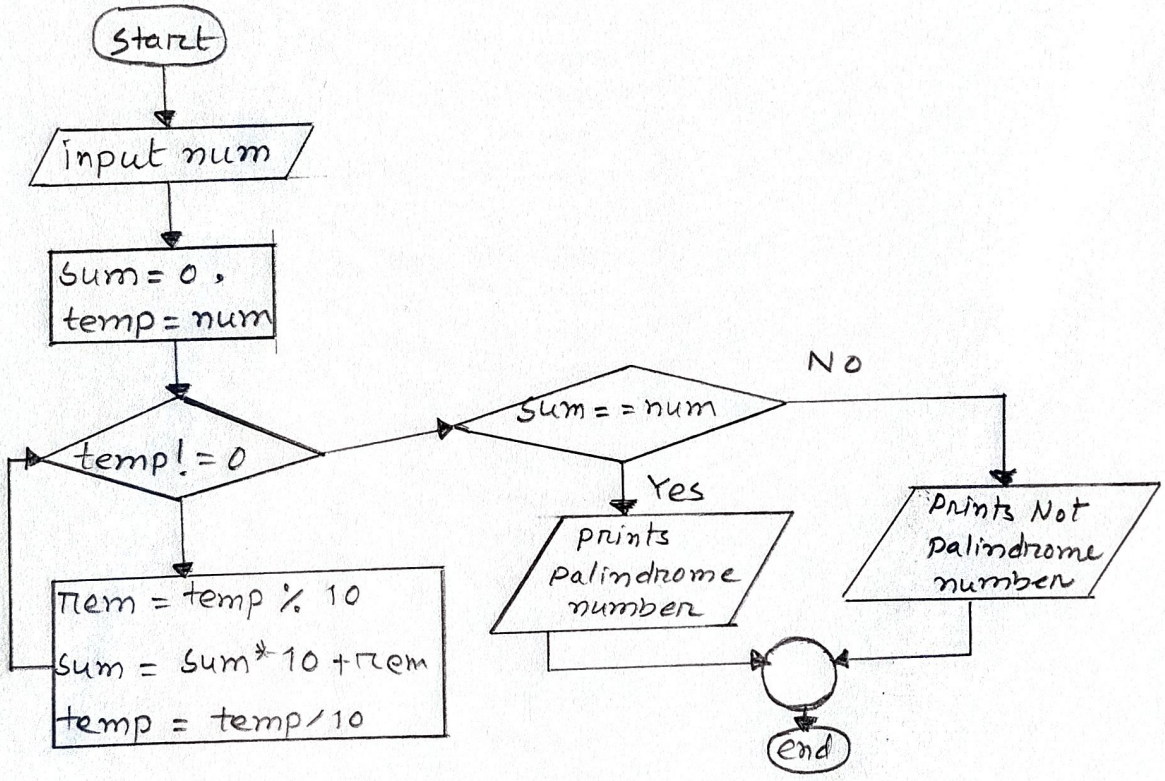
... Algorithm:

01. Start
02. Input a number
03. temp = num, sum = 0
04. While (temp != 0)
    - rem = temp % 10
    - sum = sum*10 + rem
    - temp = temp/10
05. print sum
06. end.

02. Palindrome Number

... Explanation: A palindrome number remains the same when reversed. The program reverses the number and compares it with the original.

... Flowchart:



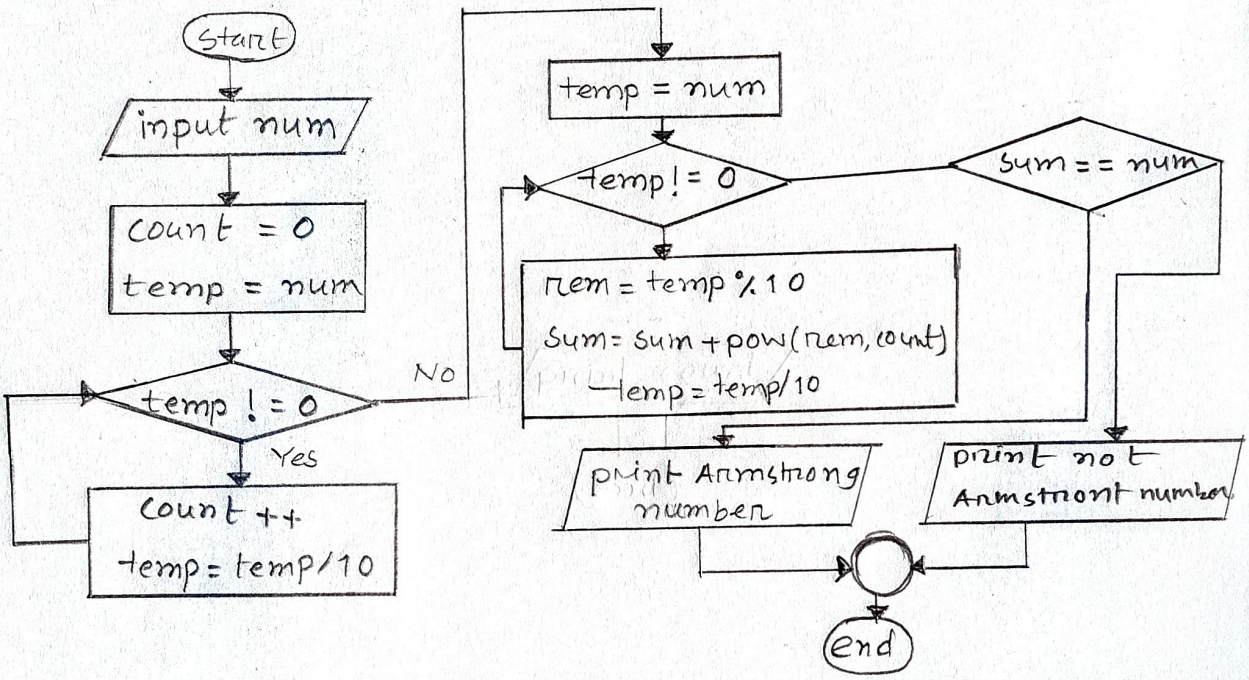
... Algorithm:

01. Start
02. Input a number
03. temp = num , sum = 0
04. while (temp != 0):
    - Rem = temp % 10
    - Sum = Sum * 10 + Rem
    - temp = temp / 10
05. If (num == sum)
    - Yes, prints it is a palindrome number
    - No, prints it is not a palindrome number
06. End

03. Armstrong Number

••• Explanation : A number is an Armstrong number if the sum of its digits raised to the power of the number of digits equals the number itself.

••• Flowchart :



••• Algorithm. :

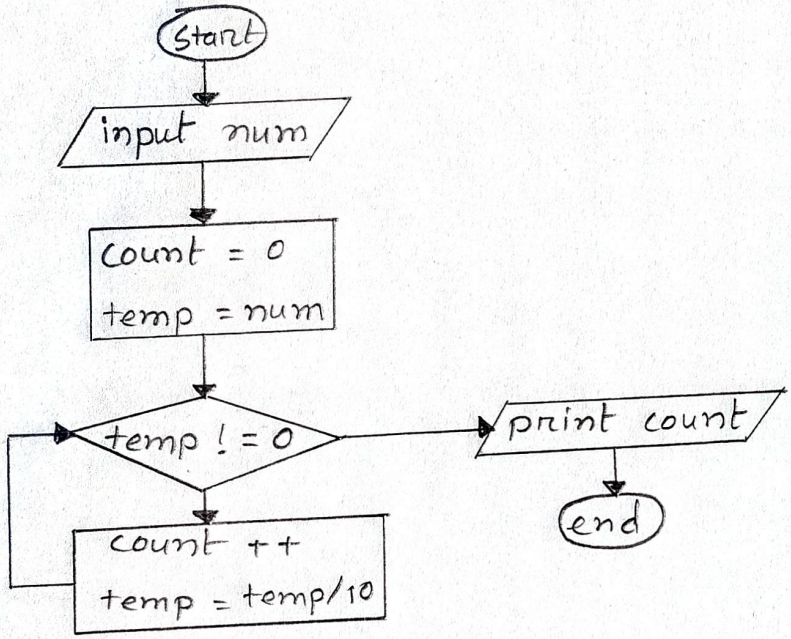
01. start
02. Input a number
03. temp = num , count = 0
04. while (temp != 0) :
    - count ++
    - temp = temp / 10
05. temp = num
06. while (temp != 0)
    - rem = temp % 10
    - sum = sum + pow(rem, count)
    - temp = temp / 10
07. Is (num == sum)
    - Yes, print it is an Armstrong number
    - No, print it is not an Armstrong number.

08. end.

04. Digit Count

... Explanation: The program counts the digits of a number by repeatedly dividing the number by 10 until it becomes zero.

... Flowchart :



... Algorithm :

01. Start
02. Input a number
03. temp = num, count = 0
04. While (temp != 0)
    - count ++
    - temp = temp / 10
05. print count
06. end

Discussion : In this section we will learn about the presentation of code and output of code. The shown code will be as in IDE and Output will be as in console.

01. Reverse on Integer

... Code :

```
#include <stdio.h>
int main (){
   int num, sum=0, temp, rem;
   printf(" Enter a number : ");
   scanf(" %d ", &num);
   temp = num;
   while ( temp != 0) {
   rem = temp % 10;
   sum = sum*10 +rem;
   temp = temp/10;
   }
   printf(" The reverse of the integer %d is %d \n",
          num , sum);
   }
```

... Output :

```
Enter a number : 123
The reverse of the integer 123 is 321
```

02. Palindrome number :

... code :

```
#include < stdio.h>
int main (){
int num, sum = 0 , temp, rem;
```

```
    printf ("Enter a number : ");
    scanf("%d", &num);
    temp = num;
    while (temp != 0)
    {
      rem = temp % 10;
      sum = sum * 10 + rem;
      temp = temp / 10;
    }
    if (sum == num)
    {
      printf ("%d is a palindrome number :", num);
    }
    else
    {
      printf("%d is not a palindrome number. ", num);
    }
    return 0;
}
```

... Output:

```
Enter a number : 121
121 is a palindrome number.
```

03. Armstrong number:

... Code:

```
#include <stdio.h>
#include<math.h>
int main(){
    int num, sum = 0, temp, rem, count=0;
    printf("Enter a number : ");
    scanf("%d", &num);
```

```
    temp = num;
    while (temp!=0)
    {
      count ++;
      temp = temp /10;
    }
    printf("%d \n", count);
    temp = num;
    while (temp != 0)
    {
      rem = temp % 10;
      sum = sum + pow(rem,count);
      temp = temp /10;
    }
    if (sum == num)
    {
      printf(" %d is an armstrong number ", num);
    }
    else
    {
      printf(" %d is not an armstrong number", num);
    }
    return 0;
}
```

... output :

```
Enter a number : 1634
4
1634 is an armstrong number.
```

04. Digit count:

... code:

```
#include <stdio.h>
int main()}
int num, count=0, temp;
 printf("Enter a number : ");
 scanf(" %d ", &num);
 temp = num;
 while (temp!=0)
 }
   count ++;
   temp = temp / 10;
 }
 printf("The total digit in this number %d is %d \n",
          num, count);

 }
```

... output:

```
Enter a number : 123
The total digits in this number 123 is 3
```

Conclusion: These programs demonstrate key programming concepts like loops, conditionals and arithmetic operations to solve number manipulation problems. The reverse and palindrome programs are practical for data validation, while the armstrong and digit count programs have applications in mathematical problem-solving. These foundational programs can be further enhanced to handle more complex seenarios efficiently.

References :

01. c standard library documentation
02. github.com
03. freecode camp.org.
04. class notes and lectures.