

RAJALAKSHMI ENGINEERING COLLEGE

An Autonomous Institution, Affiliated to Anna University
Rajalakshmi Nagar, Thandalam - 602 105



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GE23131 - PROGRAMMING USING C
(Regulation 2023)

LABORATORY MANUAL

Name G.RITHIKA A
Register No. 241801229
Year / Branch / Section I / B.Tech / AIDS 'D'
Semester I
Academic Year 2024-2025

Algorithm & Flowchart

Ex. No.:

Date:

Calculate Area and Perimeter

Write an Algorithm and draw a Flowchart to Calculate the area and perimeter of a square.

Algorithm:

Step 1 : Start

Step 2 : Declare and read a variable 'a' which is the side of square.

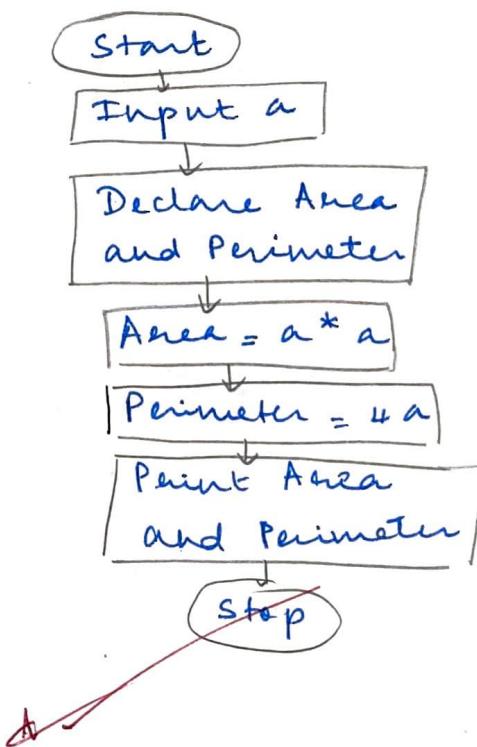
Step 3 : Declare variable 'Area' and 'Perimeter'.

Step 4 : Calculate Area = $a \times a$ and Perimeter = $4a$

Step 5 : Print area and perimeter

Step 6 : End

Flowchart:



Ex. No.:

Date:

Days to Year Conversion

Write an Algorithm and draw a Flowchart to convert the given days into years & months.

Algorithm:

Step 1: Start

Step 2: Get input days from user and store as days. Initialise years, months, days to 0.

Step 3: Start loop till $\text{days} \geq 365$.

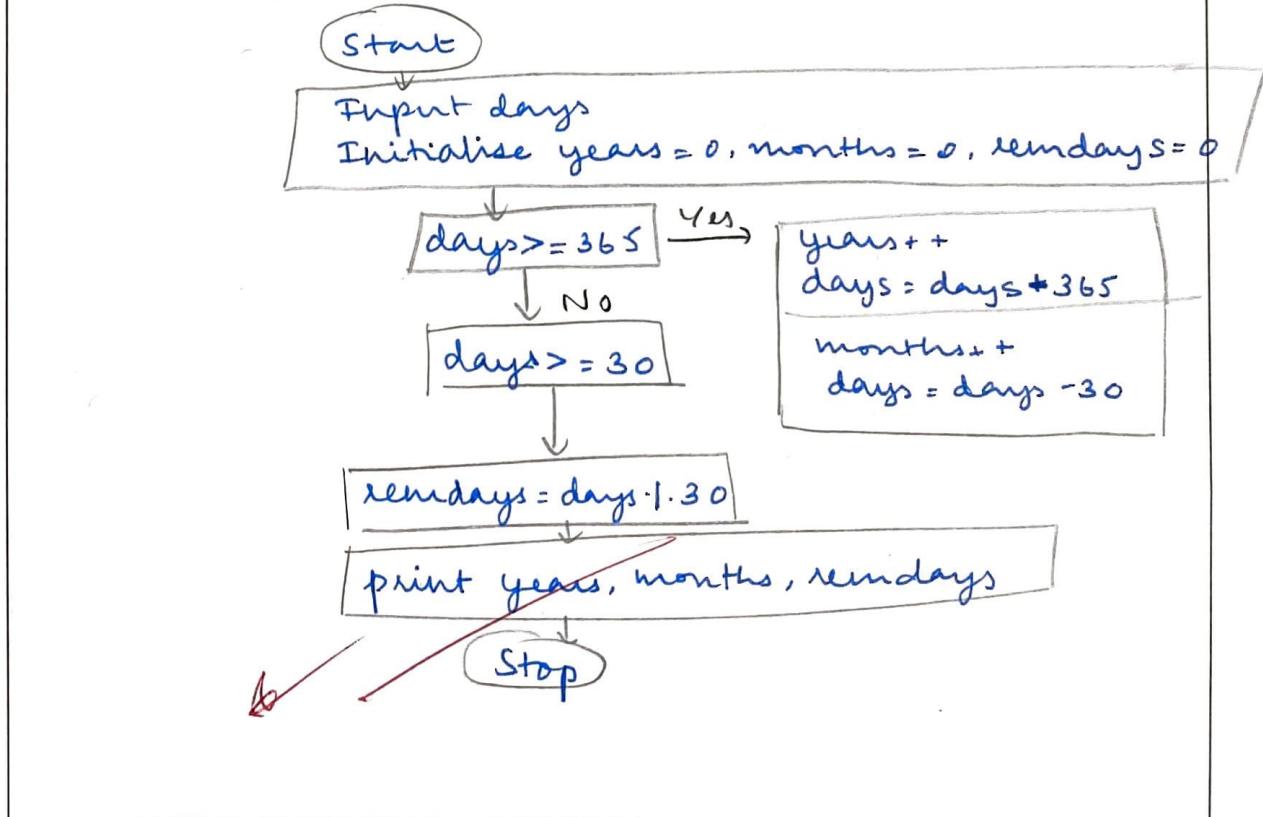
Step 4: Increment years by 1. Subtract 30 from days.

Step 5: Start loop till $\text{days} \geq 30$.

Step 6: Increment months by 1 & subtract 30 from days.

Step 7: Print years and months and remainder.

Flowchart: Step 8 : End.



Ex. No.:

Date:

Prime Number

Write an Algorithm and draw a Flowchart to check whether the given number is Prime or not.

Algorithm:

Step 1: Start

Step 2: Initialise a variable temp = 0

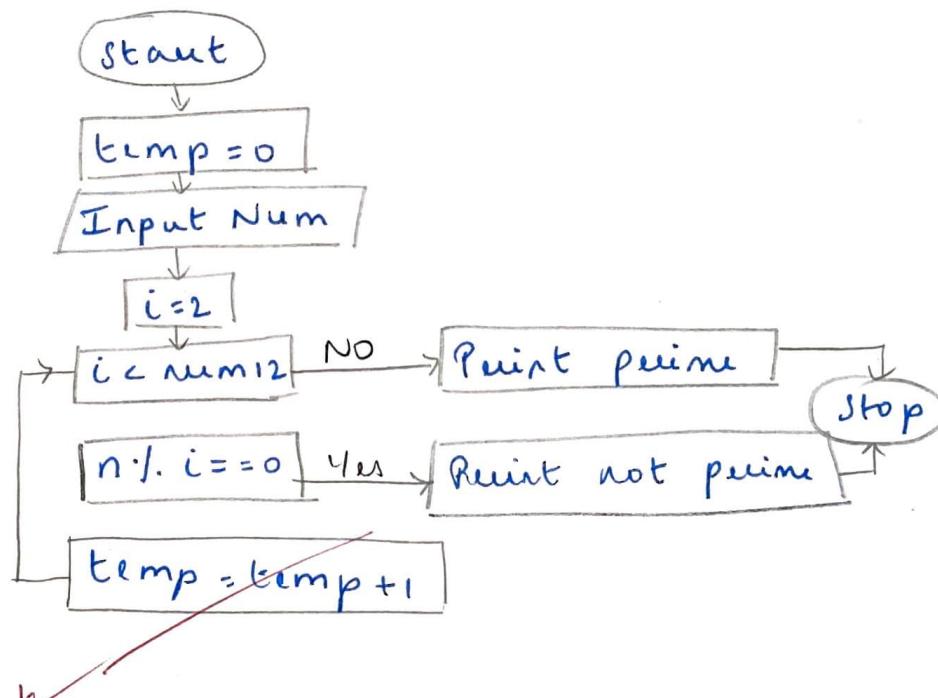
Step 3: Take a num as input

Step 4: Initialise the iteration variable loop to 1.

Step 5: Iterate a 'while' with the condition,
 $\text{loop} \neq \text{num}/2$.

Step 6: If num is divisible by loop iteration

Step 7: End.

Flowchart:

Ex. No.:

Date:

Leap Year

Write an Algorithm and draw a Flowchart to check whether the given year is Leap year or not.

Algorithm:

Step 1 : Start

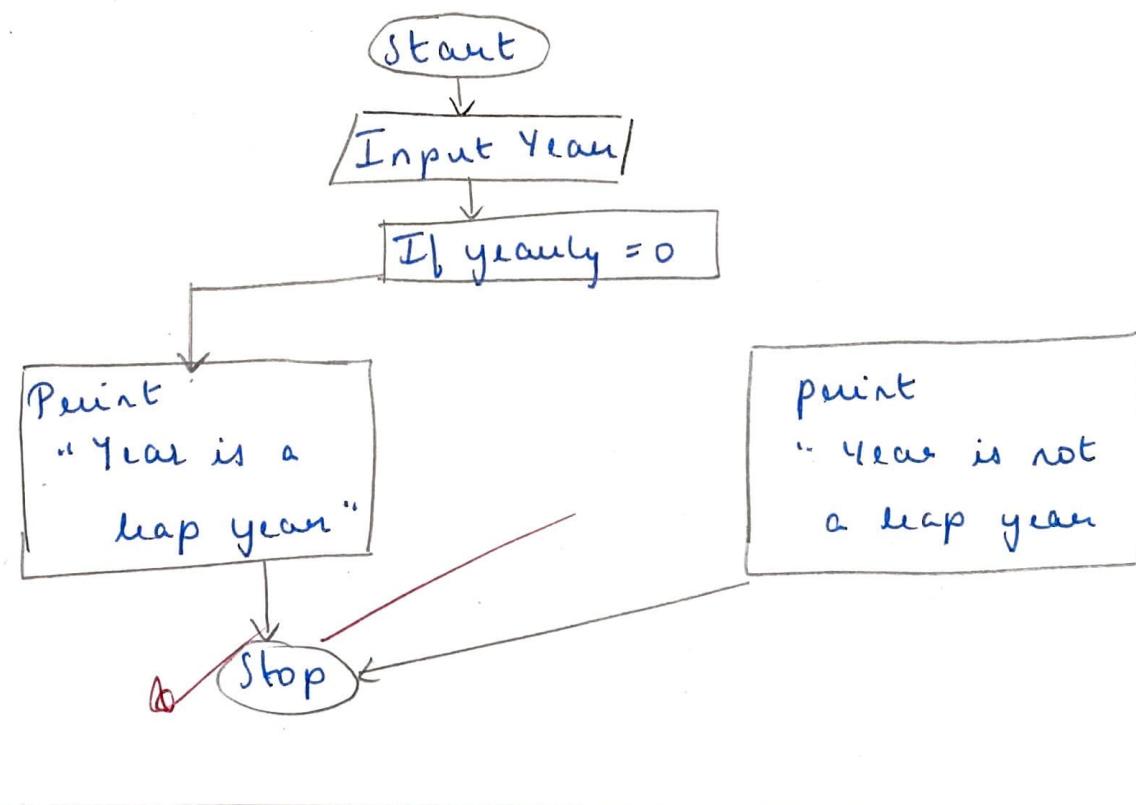
Step 2 : Input year

Step 3 : Calculate $\text{leap} = \text{Yearly} \% 4$, if remainder is zero go to step 4 otherwise step 5.

Step 4 : Print "Year is leap year".

Step 5 : Print "Year is not a leap year."

Step 6 : Stop

Flowchart:

Ex. No.:

Date:

Palindrome Number

Write an Algorithm and draw a Flowchart to check whether the given number is palindrome number or not.

Algorithm:

Step 1: Start

Step 2: Read an input as number.

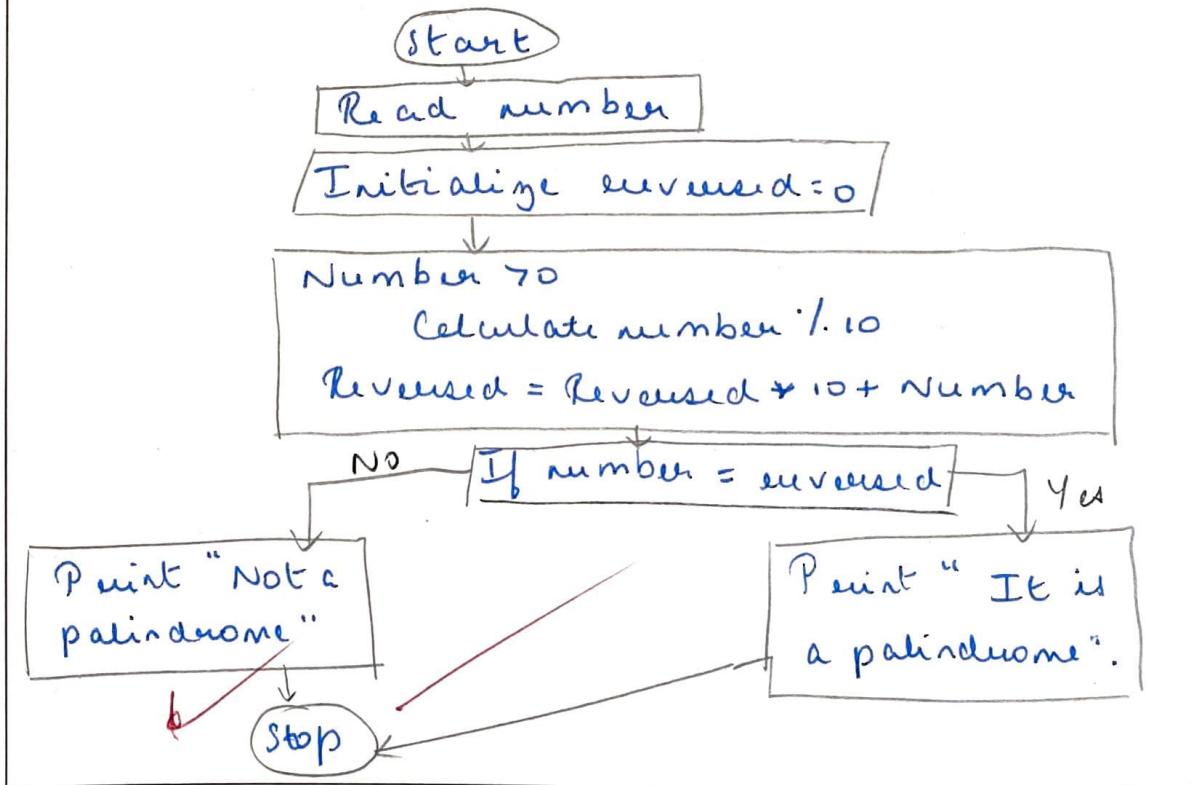
Step 3: Initialize reversed = 0.

Step 4: When number > 0; calculate number / 10
and add last digit to the reversed.

Step 5: If reversed = number, go to step 6, else
step 7.

Step 6: Print " It is not a palindrome.

Flowchart: Step 7: End.



Ex. No.:

Date:

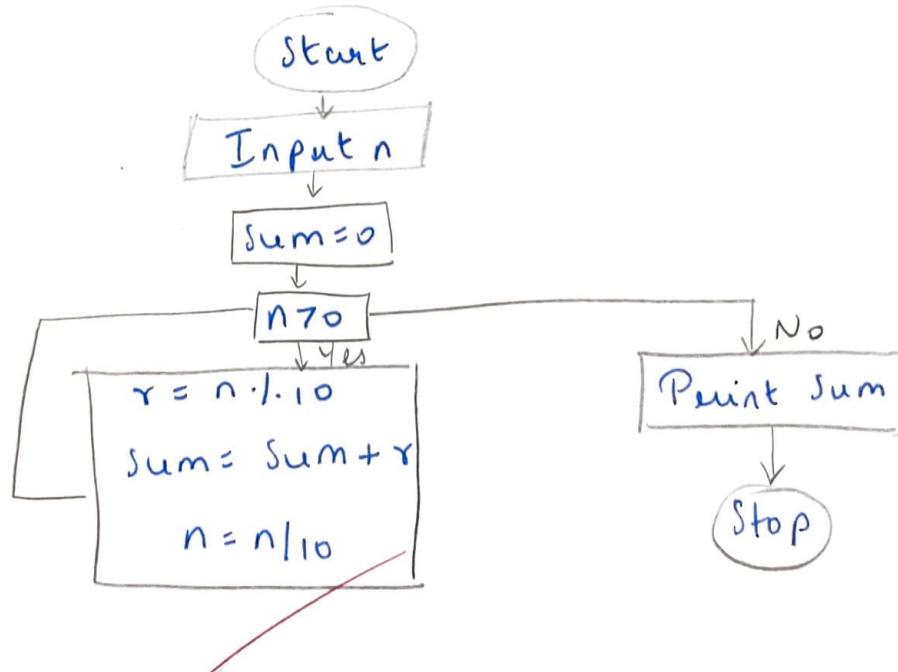
Sum of Digits

Write an Algorithm and draw a Flowchart to calculate the sum of digits in the given number.

Algorithm:

- Step 1: Start
- Step 2: Get input from user and store it in n.
- Step 3: Get loop started till $n \geq 0$.
- Step 4: Find $n \% 10$ and store the remainder in r.
- Step 5: Sum the remainder to output variable sum.
- Step 6: Divide $n / 10$.
- Step 7: End loop for $n = 0$.
- Step 8: Print the sum value.

Flowchart: Step 9: End.



Overview of C, Constants, Variables and Data Types

Ex. No.:

Date:

Say "Hello, World!" With C**Problem Statement:**

This is a simple challenge to help you practice printing to stdout.

We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string Hello, World! to stdout.

Input Format

You do not need to read any input in this challenge.

Output Format

Print **Hello, World!** to stdout.

Sample Output 1

Hello, World!

Program:

```
#include <stdio.h>
int main() {
    printf("Hello, World!");
    return 0;
}
```



Ex. No.:

Date:

Playing with Characters

Problem Statement:

This challenge will help you to learn how to take a character, a string and a sentence as input in C. To take a single character **ch** as input, you can use `scanf("%c", &ch);` and `printf("%c", ch)` writes a character specified by the argument char to stdout:

```
char ch;
scanf("%c", &ch);
printf("%c", ch);
```

This piece of code prints the character **ch**. You can take a string as input in C using `scanf("%s", s)`. But it accepts string only until it finds the first space.

In order to take a line as input, you can use `scanf("%[^n] %*c", s);` where **s** is defined as chars **[MAX_LEN]** where **MAX_LEN** is the maximum size of **s**. Here, **[]** is the `scanf` set character. **^\n** stands for taking input until a newline isn't encountered. Then, with this **%*c**, it reads the newline character and here, the used ***** indicates that this newline character is discarded.

Note: After inputting the character and the string, inputting the sentence by the above mentioned statement won't work. This is because, at the end of each line, a new line character(**\n**) is present. So, the statement: `scanf("%[^n] %*c", s);` will not work because the last statement will read a newline character from the previous line. This can be handled in a variety of ways and one of them being: `scanf("\n");` before the last statement.

Task: You have to print the character, **ch**, in the first line. Then print **s** in next line. In the last line print the sentence, **sen**.

Input Format

First, take a character, **ch** as input. Then take the string, **s** as input. Lastly, take the sentence **sen** as input

Output Format

Print three lines of output. The first line prints the character, **ch**. The second line prints the string, **s**. The third line prints the sentence, **sen**.

Sample Input 1

```
C
program
Programming using C
```

Sample Output 1

```
C
program
Programming using C
```

Program:

```
#include <stdio.h>
int main() {
    char ch;
    scanf("%c", &ch);
    printf("%c", ch);
    return 0;
```

3



Ex. No.:

Date:

Sum and Difference of Two Numbers**Problem Statement:**

The fundamental data types in c are int, float and char. Today, we're discussing int and float data types.

The printf() function prints the given statement to the console. The syntax is printf("format string",argument_list);. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write %d (integer), %c (character), %s (string), %f (float) respectively.

The scanf() function reads the input data from the console. The syntax is scanf("format string",argument_list);. For ex: The scanf("%d",&number) statement reads integer number from the console and stores the given value in variable **number**.

To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where **n** and **m** are the two integers.

Task

Your task is to take two numbers of int data type, two numbers of float data type as input and output their sum:

1. Declare **4** variables: two of type int and two of type float.
2. Read **2** lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your **4** variables.
3. Use the + and - operator to perform the following operations:
 - Print the sum and difference of two int variable on a new line.
 - Print the sum and difference of two float variable rounded to one decimal place on a new line.

Input Format

The first line contains two integers. The second line contains two floating point numbers.

Constraints: $1 \leq \text{integer variables} \leq 10^4$, $1 \leq \text{float variables} \leq 10^4$

Output Format

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to **1** decimal place) separated by a space on the second line.

Sample Input

```
10 4
4.0 2.0
```

Sample Output

```
14 6
6.0 2.0
```

Program:

```
#include <stdio.h>
int main() {
    int a, b;
    float c, d;
    scanf("%d%d", &a, &b);
    scanf("%f%f", &c, &d);
    printf("%d%d", a+b, a-b);
    printf("%d%d", c+d, c-d);
    return 0;
}
```



Ex. No.:**Date:****Average Marks****Problem Statement**

Write a program to input a name (as a single character) and marks of three tests as m₁, m₂, and m₃ of a student considering all the three marks have been given in integer format.

Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

Input Format :

Line 1 : Name(Single character)

Line 2 : Marks scored in the 3 tests separated by single space.

Output Format:

First line of output prints the name of the student. Second line of the output prints the average mark.

Constraints

Marks for each student lie in the range 0 to 100 (both inclusive)

Sample Input 1 :

A

3 4 6

Sample Output 1 :

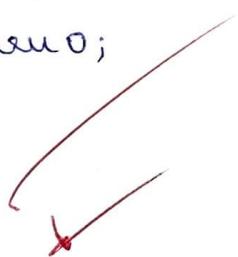
A

4

Program:

```
#include <stdio.h>

int main() {
    char name;
    int m1, m2, m3;
    int average;
    scanf ("%c", &name);
    scanf ("%d %d %d", &m1, &m2, &m3);
    average = (m1 + m2 + m3) / 3;
    printf ("%c\n", name);
    printf ("%d\n", average);
    return 0;
}
```



Ex. No.:

Date:

Basic Data Types**Problem Statement:**

Some C data types, their format specifiers, and their most common bit widths are as follows:

- *Int* ("%d"): 32 Bit integer
- *Long* ("%ld"): 64 bit integer
- *Char* ("%c"): Character type
- *Float* ("%f"): 32 bit real value
- *Double* ("%lf"): 64 bit real value

Reading

To read a data type, use the following syntax: `scanf(``formatSpecifier`", &val)`

For example, to read a *character* followed by a *double*: `char ch;`

```
double d;
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

Printing

To print a data type, use the following syntax: `printf(``formatSpecifier`", val)`

For example, to print a *character* followed by a *double*: `char ch = 'd';`

```
double d = 234.432;
printf("%c %lf", ch, d);
```

Note: You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

Input Format

Input consists of the following space-separated values: *int*, *long*, *char*, *float*, and *double*, respectively.

Output Format

Print each element on a new line in the same order it was received as input. Note that the floating-point value should be correct up to 3 decimal places and the double to 9 decimal places.

Sample Input

```
3
12345678912345
a
334.23
14049.30493
```

Sample Output

```
3
12345678912345
a
334.230
14049.304930000
```

Program:

```
#include <stdio.h>
int main() {
    int a;
    long b;
    char c;
    float d;
    double e;
    scanf ("%d %ld %f %lf", &a, &b, &c, &d, &e);
    printf ("%d\n", a);
    printf ("%ld\n", b);
    printf ("%c\n", c);
    printf ("%f\n", d);
    printf ("%lf\n", e);
    return 0;
}
```

✓

Ex. No.:

Date:

ASCII Value and Adjacent Characters

Problem Statement:

Write a program to print the ASCII value and the two adjacent characters of the given character.

Input Format: Reads the character

Output Format: First line prints the ascii value, second line prints the previous character and next character of the input character

Sample Input 1:

E

Sample Output 1:

69
D F

Program:

```
#include <stdio.h>

int main() {
    char ch;
    scanf("%c", &ch);
    int a = (int) ch;
    printf("%d\n", a);
    char b = ch - 1;
    char c = ch + 1;
    printf("%c %c\n", b, c);
    return 0;
}
```

Operators and Expressions, Managing Input and Output Operations

Ex. No.:**Date:**

Height Units

Problem Statement:

Many people think about their height in feet and inches, even in some countries that primarily use the metric system. Write a program that reads a number of feet from the user, followed by a number of inches. Once these values are read, your program should compute and display the equivalent number of centimeters.

Hint: One foot is 12 inches. One inch is 2.54 centimeters.

Input Format

First line, read the number of feet.

Second line, read the number of inches.

Output Format

In one line print the height in centimeters.

Note: All of the values should be displayed using two decimal places.

Sample Input 1

5

6

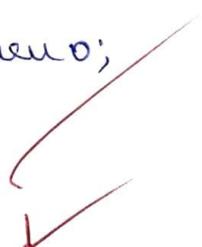
Sample Output 1

167.64

Program:

```
#include <stdio.h>

int main()
{
    int a;
    int b;
    scanf("%d", &a);
    scanf("%d", &b);
    printf("%.2f", (a * 12 * 2.54) + (b * 2.54));
    return 0;
}
```



Ex. No.:**Date:**

Arithmetic

Problem Statement:

Create a program that reads two integers, a and b, from the user. Your program should compute and display:

- The sum of a and b
- The difference when b is subtracted from a
- The product of a and b
- The quotient when a is divided by b
- The remainder when a is divided by b

Input Format

First line, read the first number.

Second line, read the second number.

Output Format

First line, print the sum of a and b

Second line, print the difference when b is subtracted from a

Third line, print the product of a and b

Fourth line, print the quotient when a is divided by b

Fifth line, print the remainder when a is divided by b

Sample Input 1

100

6

Sample Output1

106

94

600

16

4

Program:

```
#include <stdio.h>

int main()
{
    int a,b;
    scanf ("%d%d", &a, &b);
    printf ("%d\n", a+b);
    printf ("%d\n", a-b);
    printf ("%d\n", a*b);
    printf ("%d\n", a/b);
    printf ("%d\n", a%b);
    return 0;
}
```

↓