

L.T.J.S.S's

## **LOKMANYA TILAK COLLEGE OF ENGINEERING**

Sector No. 4., Vikas Nagar, Koparkhairane, Navi Mumbai -400 709.



## **FIRST YEAR ENGINEERING**



**NAME:** *Prathamesh Shivaji Chikankar*

**ROLL NO./DIV.:** *AIMLD08/D*

**BRANCH:** CSE(AI&ML)

**COURSE NAME:** *C Programming*

**COURSE CODE:** *FEL204*

**ACADEMIC YEAR:** *2020-21*                   **SEMESTER:** *II*

L.T.J.S.S's

## **LOKMANYA TILAK COLLEGE OF ENGINEERING**

Sector No. 4., Vikas Nagar, Koparkhairane, Navi Mumbai -400 709.



### **CERTIFICATE**

This is to certify that Mr./Ms. CHIKANKAR PRATHAMESH SHIVAJI of Semester II Division D bearing Roll No. AIMLD08 in the **First Year Engineering** has successfully performed 10 Experiments and 02 Assignments in the Subject **C Programming** during the period May2021 to July 2021.

**Subject Teacher**

**Head of the Department**

**Principal**

Date: 24/07/2021

## **LOKMANYA TILAK COLLEGE OF ENGINEERING**

Sector No. 4., Vikas Nagar, Koparkhairane, Navi Mumbai -400 709.



### **VISION:**

*To create technically competent and ethically responsible professionals capable of providing efficient solutions to the contemporary world.*

### **MISSION:**

*We aim to excel in our continual efforts, towards being one of the most recognized institutions by:*

- Providing a conducive environment comprising high end infrastructure and state-of-the-art laboratory facilities wherein the students, faculty and staff can collectively enhance their technical potential.*
- Encouraging innovation through research activities for the benefits of society.*
- Developing competent professionals responsive to changing technology.*

**LOKMANYA TILAK COLLEGE OF ENGINEERING**

Sector No. 4., Vikas Nagar, Koparkhairane, Navi Mumbai -400 709.

**Course Outcomes (COs)**

Academic Year	Class / Semester	Subject Name	Subject Code
2020-21	F.E/ II	C Programming	FEL204

C01	Formulate simple algorithms for arithmetic, logical problems and translate them to programs in C language
CO2	Implement, test and execute programs comprising of control structures
CO3	Decompose a problem into functions and synthesize a complete program.
C04	Demonstrate the use of arrays, strings and structures in C language.
CO5	Understand the concept of pointers

**Student Signature:** Prathu**Student Name:** Prathamesh Chikankar

**LABORATORY ASSESSMENT**

Student's Name	Prathamesh Chikankar	Div & Roll No	D & 08
Class / Sem.	F.E. - II	Academic Year	2020-21
Name of the Subject	C Programming	Course Code	FEL204

**Assessment Parameters for Experiments**

Title (Experiments)	Date of performance	Date Of submission
1 Study of basic data types and I/O operations.	01/06/2021	10/06/2021
2 Study of branching statements.	08/06/2021	14/06/2021
3 Study of looping statements.	21/06/2021	05/07/2021
4 Study of functions.	01/07/2021	05/07/2021
5 Study of recursion(function).	02/07/2021	09/07/2021
6 Study of Arrays.	09/07/2021	22/07/2021
7 Study of strings.	12/07/2021	22/07/2021
8 Study of structures.	15/07/2021	22/07/2021
9 Study of pointers.	01/07/2021	22/07/2021
10 Study of sum of series and pattern programs.	23/06/2021	22/07/2021
Assignment no-1		23/07/2021
Assignment no-02		23/07/2021

Student Signature

Subject Teacher Signature



# Experiment No: 1

## **Aim:**

**Study of basic data types and I/O operations.**

Name and roll no of student	Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	D	<i>01/06/2021</i>	<i>10/06/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

### **Program list:**

- 1. Write a program to find the simple interest for the Amount (P), Rate of Interest (R) and Number of years (N).**
- 2. Write a program to convert the given temperature in degree centigrade to Fahrenheit and vice versa.**
- 3. Write a program to obtain roots of second order quadratic equation of the form  $ax^2+bx+c=0$  where a, b, c is non-zero.**
- 4. Write a program to swap two numbers without using third variable.**

Program 1 :

Problem :

Write a program to find the Simple Interest for the Amount (P), Rate of Interest (R) & No. of years (N).

Algorithm :

Step 0 : START

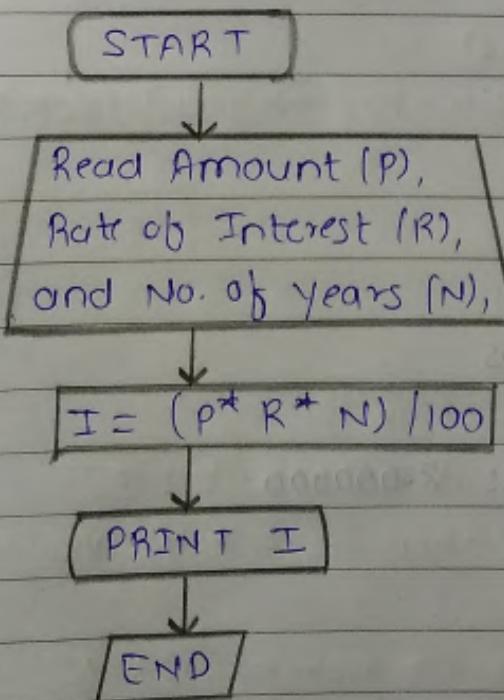
Step 1 : Read Amount (P), Rate of Interest (R) and Number of years (N)

Step 2 : Convert P, R and N into Simple Interest (I)  
using the formula  $I = (P * R * N) / 100$

Step 3 : Display the result

Step 4 : STOP

Flowchart :



PAGE No.	02
DATE	/ /

### Program:

```
#include <stdio.h>
main()
{
    float P, R, N, I;
    printf("Enter the amount \n");
    scanf("%f", &P);
    printf("Enter the rate \n");
    scanf("%f", &R);
    printf("Enter the no. of years \n");
    scanf("%f", &N);
    I = (P * R * N) / 100;
    printf("The simple interest is %f \n", I);
```

### Output :

Enter the amount

100

Enter the rate

4

Enter the no. of years

2

The simple interest is 8.000000

Program 2:

Problem:

write a program to convert the given temperature in degree centigrade to fahrenheit & vice versa.

Algorithm:

Step 0: START

Step 1: Read the temperature in degree centigrade

Step 2: Convert the centigrade to fahrenheit using the formula  $F = (9/5 * C) + 32$ .

Step 3: print F.

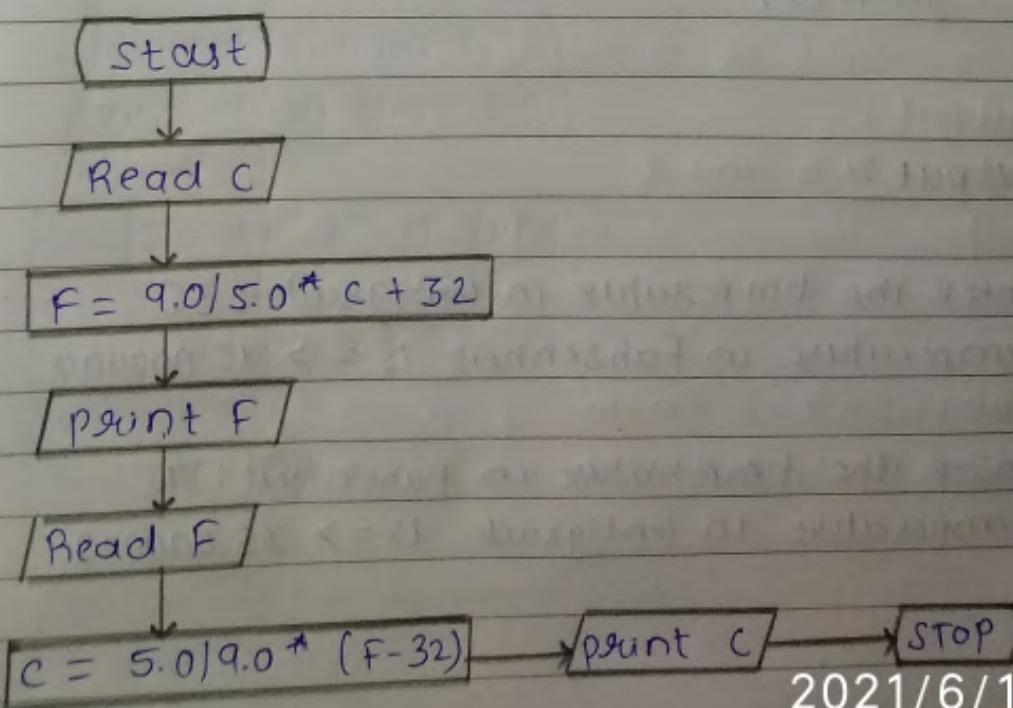
Step 4: Read the temperature in degree fahrenheit.

Step 5: Convert the fahrenheit to centigrade using the formula  $C = 5/9 * (F - 32)$

Step 6: print c.

Step 7: STOP.

Flowchart:



program:

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    float c,f;
    printf("Enter the temperature in centigrade:");
    scanf("%f",&c);
    f = (9.0/5.0 * c) + 32;
    printf("Temperature in Fahrenheit is =>%f\n",f);

    printf("Now Enter the temperature in Fahrenheit:");
    scanf("%f",&f);
    c = 5.0/9.0 * (f - 32);
    printf("Temperature in centigrade is =>%f\n",c);
    getch();
}
```

Output:

Enter the temperature in centigrade: 35  
 Temperature in Fahrenheit is => 95.000000

Enter the temperature in Fahrenheit: 95  
 Temperature in centigrade is => 35.000000

Program 3:

problem:

write a program to obtain roots of second order quadratic equation of the form  $ax^2 + bx + c = 0$  where  $a, b, c$  is non-zero.

Algorithm:

Step 0: START

Step 1: Read coefficients  $a, b, c$  of the equation

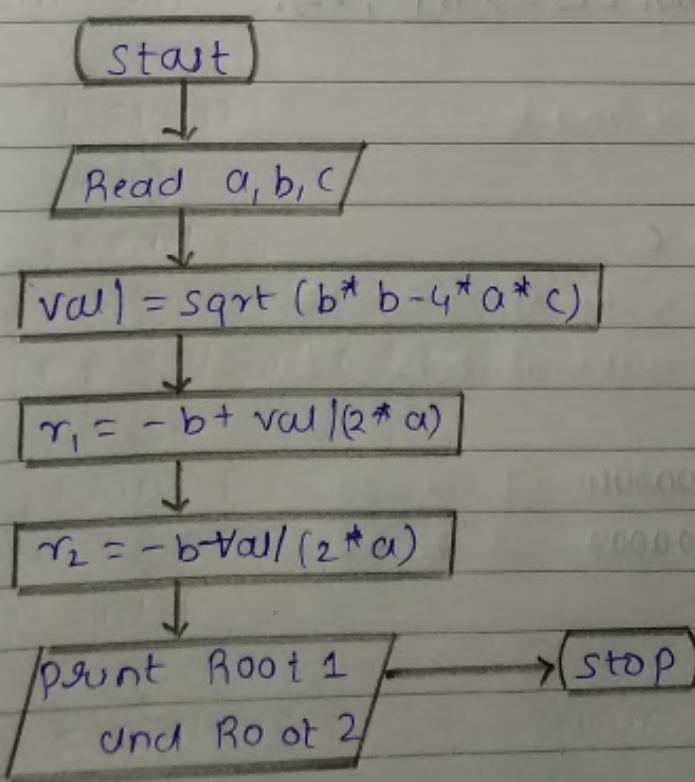
$$ax^2 + bx + c = 0.$$

Step 2: calculate roots of quadratic equation by formula  $\text{root}_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Step 3: display the results.

Step 4: STOP

Flowchart:



program:

```
#include <stdio.h>
#include <math.h>
main()
{
    int a,b,c;
    float r1,r2,Val;
    printf("Enter a,b and c\n");
    scanf("%d%d%d", &a, &b, &c);
    Val = sqrt((b*b) - (4*a*c));
    r1 = ((-b)+Val)/(2*a);
    printf("root1 is => %f\n", r1);
    r2 = ((-b)-Val)/(2*a);
    printf("root2 is =>%f", r2);
}
```

output:

Enter a,b and c

1

4

4

root1 is => -2.000000

root2 is => -2.000000

program 4:

problem:

write a program to swap two numbers without using third variable.

Algorithm:

Step 0: START

Step 1: Read two numbers a and b

Step 2: Swapping without third variable can be done by applying the following logic:

$$a = a + b$$

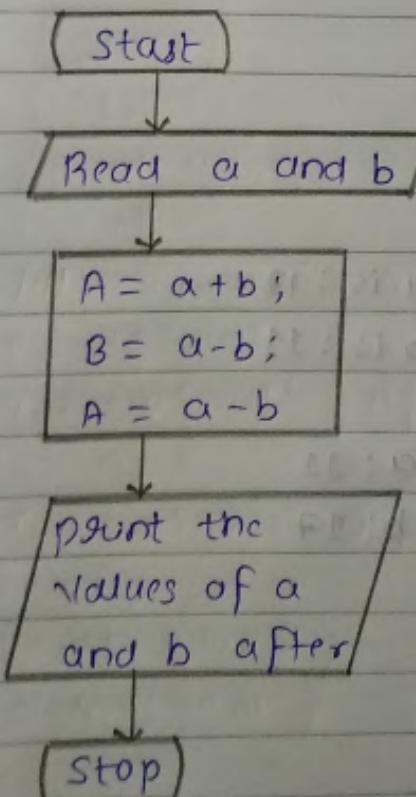
$$b = a - b$$

$$a = a - b$$

Step 3: Display the result after swapping.

Step 4: STOP

flowchart:



program:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b;
    printf("Enter value of a and b:\n");
    scanf("%d%d", &a, &b);
    printf("Before swapping value of a is: %d\n", a);
    printf("Before swapping value of b is: %d\n\n", b);
    a = a+b;
    b = a-b;
    a = a-b;
    printf("After swapping value of a: %d\n", a);
    printf("After swapping value of b: %d\n", b);
    getch();
}
```

output:

Enter value of a and b:

17

11

Before swapping value of a is : 17

Before swapping value of b is : 11

After swapping value of a: 11

After swapping value of b: 17

## **Experiment No: 2**

**Aim:**

**Study of branching statements.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Chikankar</i>	<i>CSE(AI&amp;ML) D</i>	<i>08/06/2021</i>	<i>14/06/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

**Program list:**

- 1. Write a program to find whether the given number entered is even or odd.**
- 2. Write a program to read marks of the student and display grade based on the following norms using else if Ladder:**  
>=75: Grade A  
>=60 and <75: Grade B  
>=40 and <60: Grade C  
<40: Fail
- 3. Write a program to develop a simple calculator that accepts two floating point numbers from the keyboard. Display a menu to the user and get the user's choice. Perform the operation and display the result using switch statement.**

program 1 :

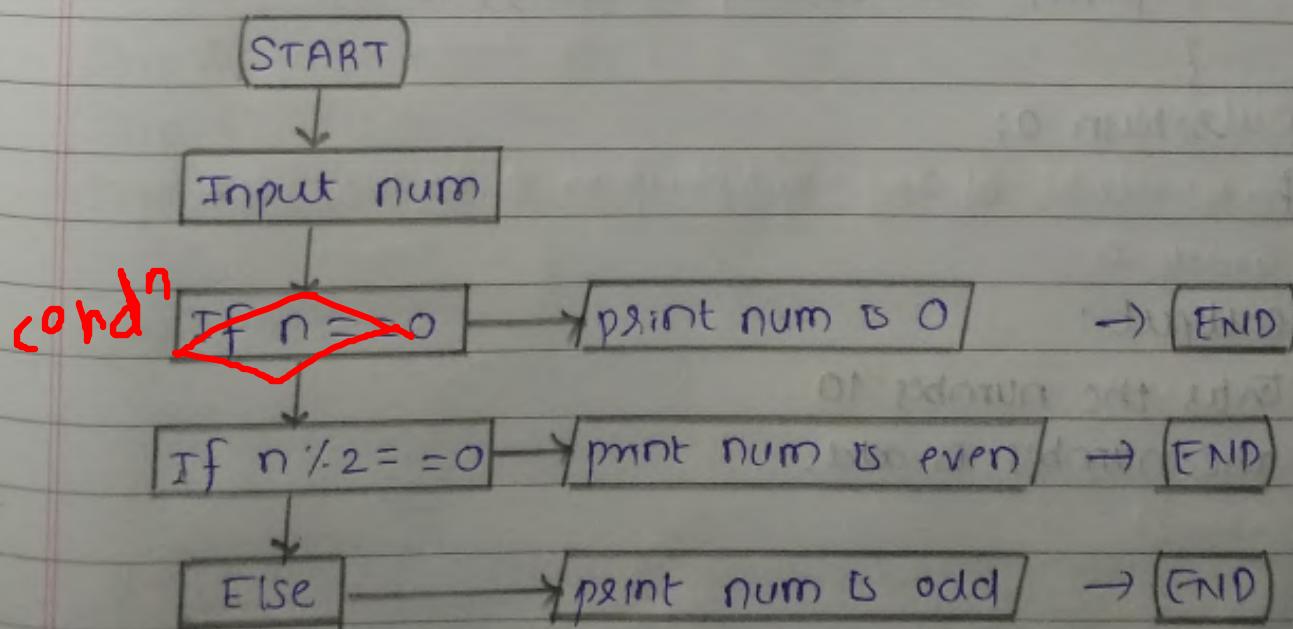
problem :

write a program to find whether the given number entered is even or odd.

Algorithm :

- Step 0: START
- Step 1: Take user input (number)
- Step 2: Check if number is equal to zero.
- Step 3: If true, print number is zero
- Step 4: If false, check number is divisible by 2.
- Step 5: If true, print number is even
- Step 6: If none of above case is true, then print number is odd.
- Step 7: Display the result
- Step 8: STOP

flowchart :



PAGE NO.	02
DATE	/ /

program :

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the number");
    scanf("%d", &n);
    if (n == 0)
    {
        printf("the given number is zero");
    }
    else if (n%2 == 0)
    {
        printf("the number is even");
    }
    else
    {
        printf("the number is odd");
    }
    return 0;
}
```

Output :

Enter the number 10

The number is even.

## Program 2

### problem 1

write a program to read marks of the student and display grade based on the following norms using else if ladder:

$\geq 75$ : Grade A

$\geq 60$  and  $< 75$ : Grade B

$\geq 40$  and  $< 60$ : Grade C

$< 40$ : fail

### Algorithm :

Step 0: START

Step 1: Take marks from the user input

Step 2: If input is greater than or equal to 75  
then print Grade A

Step 3: If input is greater than or equal to 60 &  
less than 75 then print Grade B

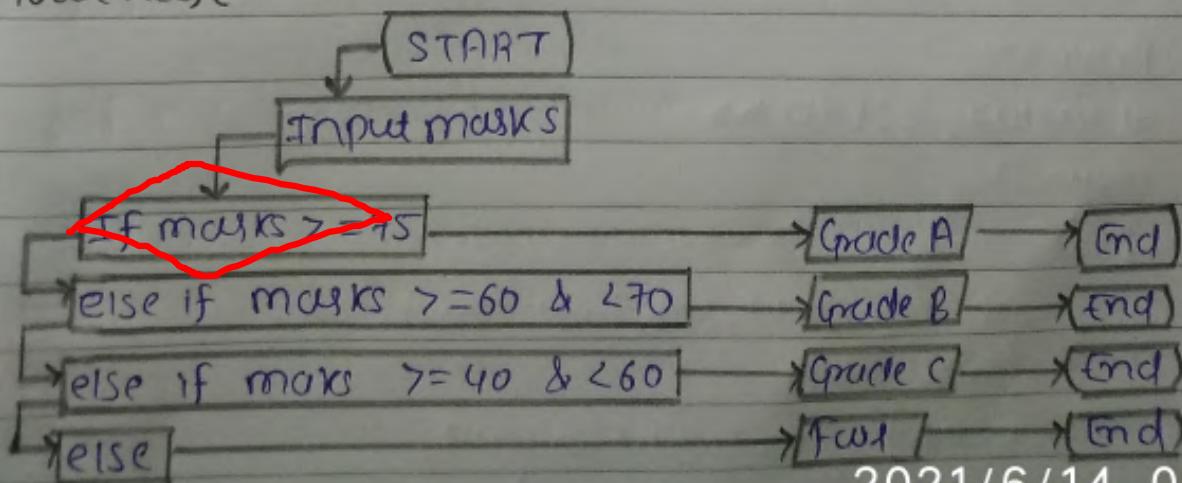
Step 4: If input is greater than or equal to 40 and  
less than 60 then print Grade C

Step 5: Else print fail (if input is less than 40)

Step 6: Display the result.

Step 7: STOP

### flowchart



PAGE NO.	04
DATE	/ /

program:

```
#include <stdio.h>
main()
{
    float marks;
    printf("Enter marks of the student from 0 to 100");
    scanf("%f", &marks);
    if(marks >= 75)
        printf("Grade A");
    else if(marks >= 60 & marks < 75)
        printf("Grade B");
    else if(marks >= 40 & marks < 60)
        printf("Grade C");
    else
        printf("Fail");
    return 0;
}
```

Output:

Enter marks of the student from 0 to 100 : 97  
Grade A

program 3:

problem:

write a program to develop a simple calculator that accepts two floating point numbers from the keyboard. Display a menu to the user and get the user's choice. Perform the operation and display the result using switch statement.

Algorithm:

Step 0: START

Step 1: Declare 2 variable in int as x and y along with add, sub, mul, choice;

Step 2: also declare float as div

Step 3: print the whole choice of operators and ask the user to input his choice.

Step 4: Switch case choice. Deduce numbers

Step 5: If user input '1' i.e. '+' addition of two input

Step 6: If user input '2' i.e. '-' subtraction of x and y & print answer.

Step 7: If user input '3' i.e. '\*' choice multiplication of x and y & will print answer

Step 8: If user input '4' i.e. '/' division of two inputs x and y & print answer.

Step 9: If user input incorrect value i.e. will print wrong choice

Step 10: Display the result.

Step 11: STOP

PAGE NO.	06
DATE	/ /

flowchart

START

Input two numbers

Input the choice of operator

If switch case 1.addition

Add two numbers

and print

End/break

If switch case 2.subtraction

Sub two numbers

End/break

and print

If switch case 3.multiplication

Mul two numbers

End/break

and print

If switch case 4.division

Div two numbers

End/break

and print

If invalid choice

Default

print wrong

choice

End/break

program:

#include &lt;stdio.h&gt;

main()

{

int x,y,add,sub,mul,choice;

float div;

printf("1. addition\n");

printf("2. subtraction\n");

```

printf("3. multiplication\n");
printf("4. Division\n");
printf("Enter numbers =>\n");
scanf("%d%d", &x, &y);
printf("Enter choice =>\n");
scanf("%d", &choice);
switch(choice)
{

```

case 1:

```

add = x + y;
printf("Addition is =>%d", add);
break;

```

case 2:

```

sub = x - y;
printf("Subtraction is =>%d", sub);
break;

```

case 3:

```

mul = x * y;
printf("Multiplication is =>%d", mul);
break;

```

case 4:

```

div = x / y;
printf("Division is =>%d", div);
break;

```

default:

```

printf("Wrong choice");
break;

```

}

return 0;

PAGE No.	08
DATE	/ /

output:

1. addition

2. subtraction

3. multiplication

4. DIVISION

Enter numbers =

97

77

Enter choice =

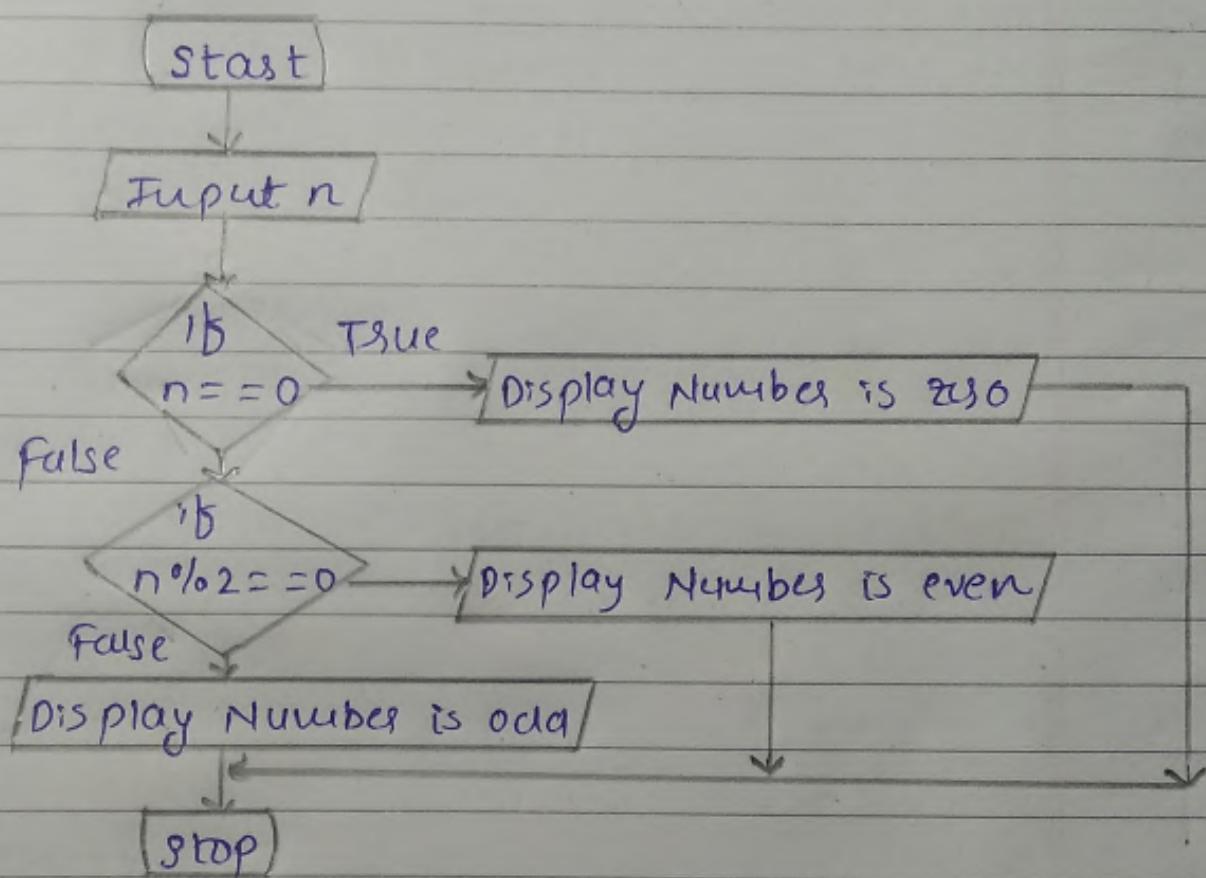
3

multiplication is  $\Rightarrow 7469$

## Experiment 02

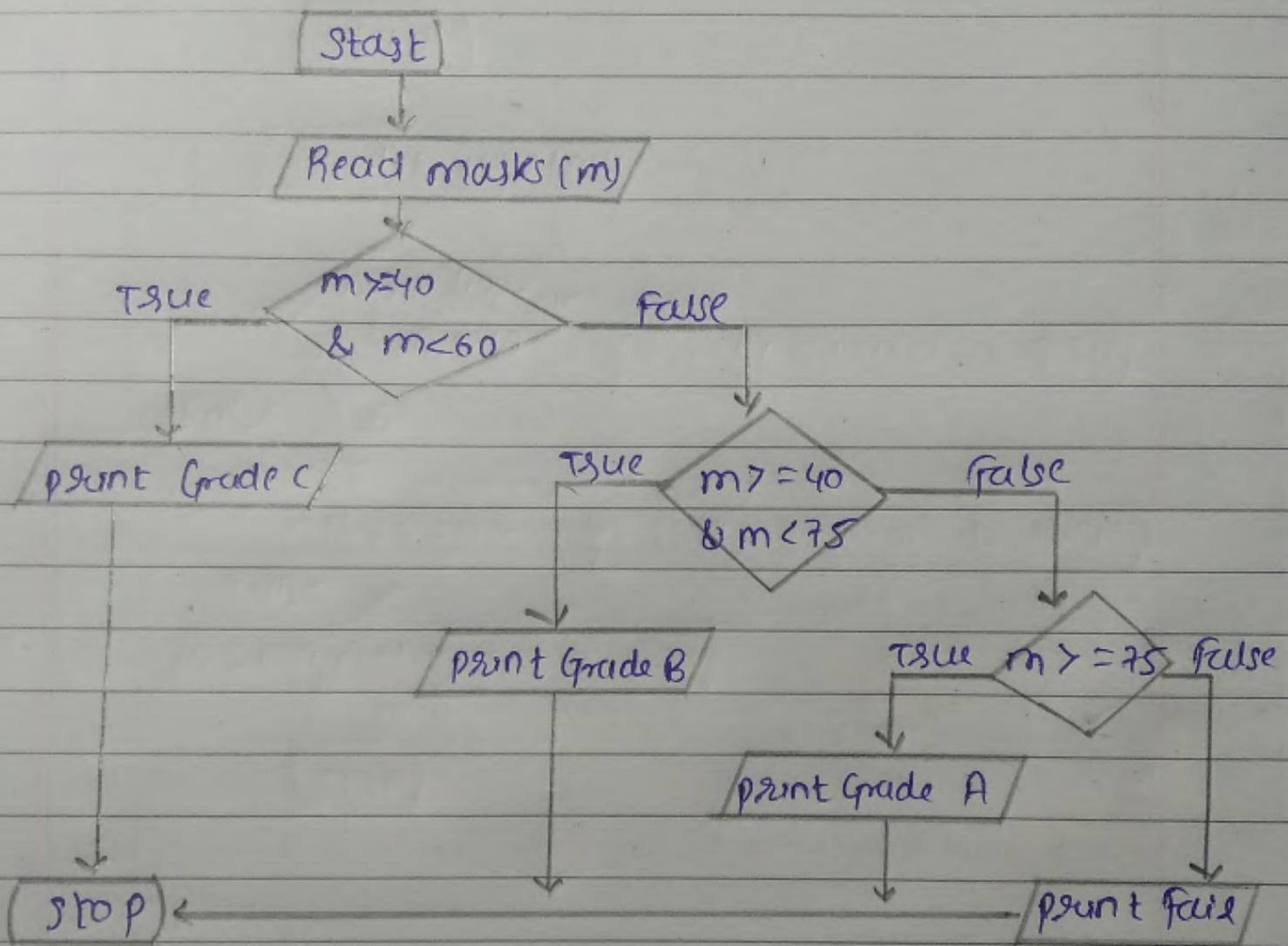
Program 1

Flowchart:



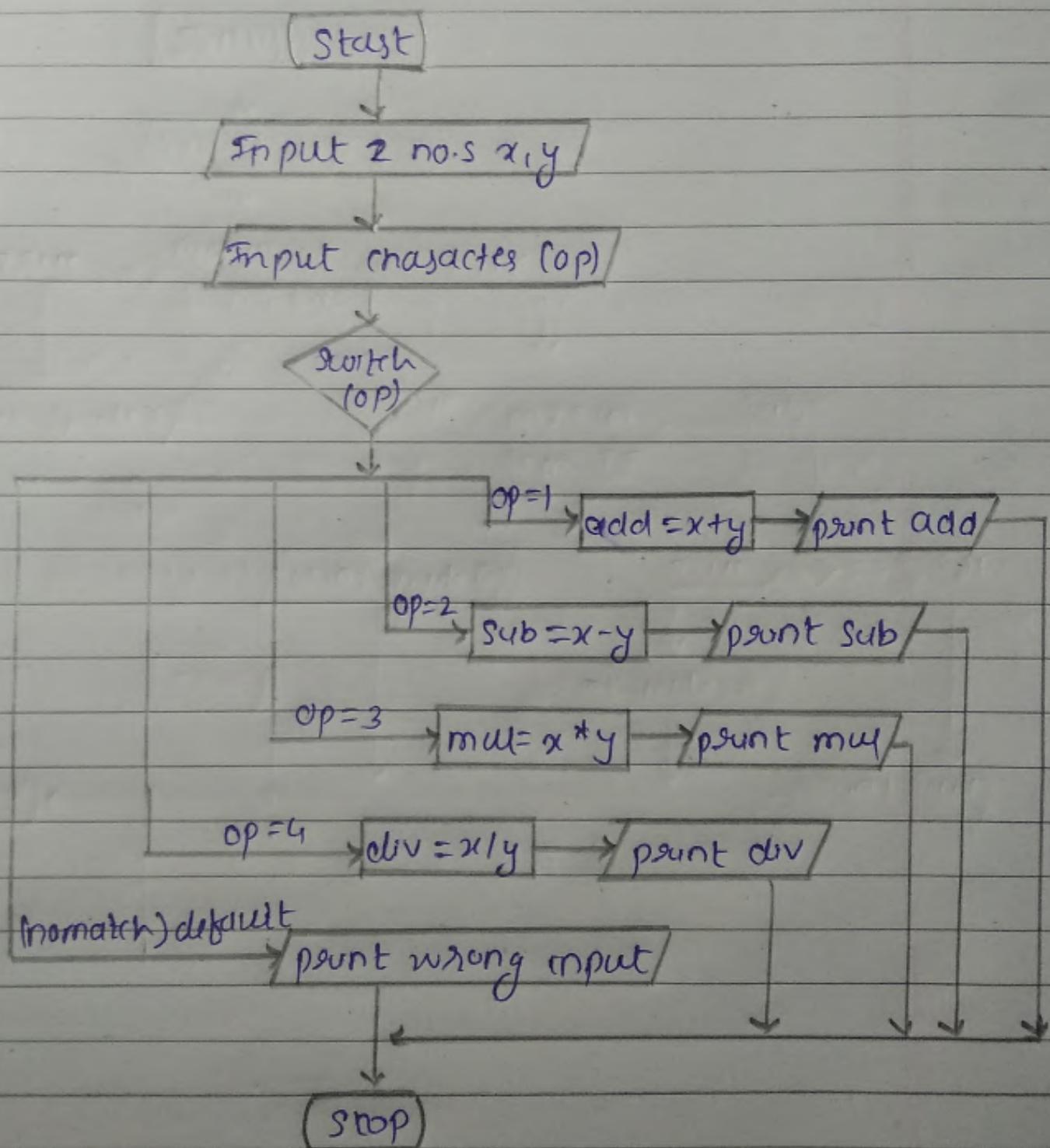
program 2

flowchart:



Programming 3

Flowchart :



# Experiment No: 3

**Aim:**

**Study of looping statements.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	<i>First year and D</i>	<i>21/06/2021</i>	<i>05/07/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

**Program list:**

- 1. Write a program to print the first ten terms of the Fibonacci Sequence assuming the first two terms as 0 and 1.**
- 2. Write a program to check whether given number is prime or not.**
- 3. Write a program to reverse a given number using while loop.**

Program 1:

problem :

write a program to print the first ten terms of the fibonacci sequence assuming the first two terms as 0 and 1.

program :

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

```
int main()
```

```
{
```

```
    int n=10;
```

```
    int firstnum = 0;
```

```
    int Secondnum = 1;
```

```
    int nextnum;
```

```
    printf("first ten terms of the fibonacci sequence");
```

```
    for(int i=0; i<n; i++)
```

```
{
```

```
        printf("%d,", firstnum);
```

```
        nextnum = firstnum+Secondnum;
```

```
        firstnum = Secondnum;
```

```
        Secondnum = nextnum;
```

```
}
```

```
    return 0;
```

```
}
```

Output :

first ten terms of the fibonacci sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

Program 2:Program:

Write a program to check whether the given number is prime or not.

Program:

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

```
main()
```

```
{
```

```
    int n, i, c = 0;
```

```
    printf("Enter any number n:");
```

```
    scanf("%d", &n);
```

```
    for (i=2; i<=n; i++)
```

```
{
```

```
        if (n % i == 0)
```

```
{
```

```
            c++;
```

```
}
```

```
}
```

```
    if (c == 2)
```

```
{
```

```
        printf("n is a prime number");
```

```
}
```

```
else
```

```
{
```

```
    printf("n is not a prime number");
```

```
}
```

```
return 0;
```

```
}
```

Output:

Enter any number n: 97

n is a prime number

program 3:

problem:

write a program to reverse a given number  
using while loop.

program:

```
#include <stdio.h>
int main ()
{
    int n, rev = 0, remainder;
    printf("Enter an integer: ");
    scanf("%d", &n);
    while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n /= 10;
    }
    printf("Reversed number = %d", rev);
    return 0;
}
```

Output:

Enter an integer: 6497

Reversed number = 7946

## Experiment No: 4

**Aim:**

**Study of functions.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	<i>First Year and D</i>	<i>01/07/2021</i>	<i>05/07/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

**Program list:**

- 1. Devise a function called min (x, y) that returns the smaller of two double values. Test the function with sample data.**
- 2. Write a program to find sum of digits of a number using function.**

Program 1:

problem:

Devise a function called min(x,y) that returns the smaller of two double values. Test the function with sample data.

Program :

```
#include<stdio.h>
#include<math.h>
int min(int x, int y)
{
    int minNumber = 0;
    if(x < y)
    {
        minNumber = x;
    }
    else
    {
        minNumber = y;
    }
    return minNumber;
}
int main()
{
    int number1;
    int number2;
    printf("Enter first number : \n");
    scanf("%d", &number1);
    printf("Enter second numbers : \n");
    scanf("%d", &number2);
```

```
int operation = min (number1, number2);
printf ("smallest number is %d", operation);
return 0;
```

{}

Output:

Enter first number:

97

Enter Second number :

77

Smallest number is 77

program 2:

problem:

write a program to find sum of digits of a number using function.

program:

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

```
#include <conio.h>
```

```
main()
```

```
{
```

```
int n, sum = 0, remainder;
```

```
printf("Enter the Number : ");
```

```
scanf("%d", &n);
```

```
while (n != 0)
```

```
{
```

```
remainder = n % 10;
```

```
sum = sum + remainder;
```

```
n = n / 10;
```

```
}
```

```
printf("sum of digits of entered Number: %d", sum);
```

```
getch();
```

```
}
```

Output:

Enter the Number: 786

Sum of digits of entered Number : 21

## Experiment No: 5

**Aim:**

**Study of recursion(function).**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> PRATHAMESH SHIVAJI CHIKANKAR	<i>FIRST YEAR AND D</i>	02/07/21	09/07/2021
<b>Roll no:</b> <i>AIMLD08</i>			

**Program list:**

- 1. Write a program to calculate the factorial of a given number using recursive function.**
- 2. Write a program to compute  $a^b$  using recursion.**

Program 1:

problem:

write a program to calculate the factorial of a given number using recursive function.

program:

```
#include <stdio.h>
int fact(int);
int main()
{
    int x,n;
    printf("Enter the Number to find factorial :");
    scanf("%d", &n);
    x = fact(n);
    printf("factorial of %d is %d", n, x);
    return 0;
}

int fact(int n)
{
    if (n == 0)
        return 1;
    return (n * fact(n-1));
}
```

Output:

Enter the Number to find factorial : 5

factorial of 5 is 120

PAGE NO.	02
DATE	/ /

Program 2 :

Problem :

Write a program to compute  $a^b$  using recursion.

Program :

```
#include <stdio.h>
int power(int n1, int n2);
int main()
{
    int base, a, result;
    printf("Enter base number: ");
    scanf("%d", &base);
    printf("Enter power number (positive integers): ");
    scanf("%d", &a);
    result = power(base, a);
    printf("%d ^ %d = %d", base, a, result);
    return 0;
}
```

```
int power(int base, int a)
```

```
{
```

```
if (a != 0)
```

```
return (base * power(base, a - 1));
```

```
else
```

```
return 1;
```

```
}
```

Output:

Enter base number : 3

Enter power number (positive integers) : 4

$3^4 = 81$

## Experiment No: 6

Aim:

**Study of Arrays.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	<i>First Year and D</i>	<i>09/07/2021</i>	<i>22/07/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

**Program list:**

- 1. Write a program to sort elements of an array (Bubble sort).**
- 2. Write a program to add two matrices using multi-dimensional arrays.**

Program 1 :problem :

write a C program to sort elements of an array (Bubble sort)

Program :

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

```
main ()
```

```
{
```

```
int a[5], i, j, t;
```

```
printf ("Enter array =>\n");
```

```
for (i = 0; i < 5; i++)
```

```
{
```

```
scanf ("%d", &a[i]);
```

```
}
```

```
for (i = 0; i < 5; i++)
```

```
{
```

```
for (j = 0; j < 5; j++)
```

```
{
```

```
if (a[j] > a[j + 1])
```

```
{
```

```
t = a[j];
```

```
a[j] = a[j + 1];
```

```
a[j + 1] = t;
```

```
}
```

```
}
```

```
printf ("Sorted array element are =>\n");
```

```
for (i = 0; i < 5; i++)
```

```
{
```

```
printf ("%d\n", a[i]);
```

```
}
```

Output:

Enter number of elements

4

Enter the elements of array

77

97

64

94

Sorted array is

64    77    94    97

Program 2:Problem:

Write a program to add two matrices using multidimensional arrays.

Program:

```
#include <stdio.h>
main ()
{
    int a[2][2], b[2][2], c[2][2], i, j;
    printf("Enter matrix A => \n");
    for (i=0; i<2; i++)
    {
        for (j=0; j<2; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter matrix B => \n");
    for (i=0; i<2; i++)
    {
        for (j=0; j<2; j++)
        {
            scanf("%d", &b[i][j]);
        }
    }
}
```

```
{  
    for (j=0; j<2; j++)  
    {  
        scanf ("%d", &b[i][j]);  
    }  
}  
  
printf ("Addition of matrix A and B is=>\n");  
for (i=0; i<2; i++)  
{  
    for (j=0; j<2; j++)  
    {  
        c[i][j]=0;  
        c[i][j]= a[i][j]+b[i][j];  
    }  
}  
  
for (i=0; i<2; i++)  
{  
    for (j=0; j<2; j++)  
    {  
        printf ("%d\t", c[i][j]);  
    }  
    printf ("\n");  
}  
}
```

### Output:

Next Page

80 QM 08  
80

ATMLD08

PAGE NO.	04
DATE	/ /

Entes matrice A =>

1

2

3

4

Entes matrice B =>

5

6

7

8

Addition of matrice A and B is =>

6

10

(A+B)ij = (Aij + Bij)

2009/10/11

## Experiment No: 7

**Aim:**

**Study of strings.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	<i>First Year and D</i>	<i>12/07/2021</i>	<i>22/07/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

**Program list:**

- 1. Write a program to find whether a given string is palindrome or not.**
- 2. Write a program to demonstrate strrev () function.**
- 3. Write a program to find frequency of characters in a string.**

Program 1:problem:

write a program to find whether a given string is palindrome or not.

Program:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char string1[120];
    int i, length;
    int flag = 0;
    printf("Enter a string:");
    scanf("%s", string1);
    length = strlen(string1);
    for(i=0; i<length; i++)
    {
        if(string1[i] != string1[length-i-1])
        {
            flag = 1;
            break;
        }
    }
    if(flag == 1)
    {
        printf("%s is not a palindrome", string1);
    }
    else
    {
        printf("%s is a palindrome", string1);
    }
    return 0;
}
```

Output:

Enter a string: IYAYI

IYAYI is a palindrome

Program 2problem:

Write a program to demonstrate `strrev()` function.

program:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[50];
    printf("The given string is = %s\n", str);
    printf("After reversing string is = %s", strrev(str));
    return 0;
}
```

Output:

Enter a string to reverse : DHRUV DP 97

Reverse of entered string is : 79 PD VURHD

Program 3problem:

Write a program to find frequency of characters in a string.

PAGE No.	03
DATE	/ /

Program :

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
main()
{
    char str[1500];
    int c=0, count[26]={0};
    printf("Enter a string :");
    gets(str);
    while (str[c] != '\0')
    {
        if (str[c] >= 'a' & & str[c] <= 'z')
            count[str[c] - 'a']++;
        c++;
    }
    for (c=0; c<26; c++)
    {
        if (count[c] != 0)
            printf("%c occurs %d times in the string.\n",
                   c + 'a', count[c]);
    }
    getch();
}

```

Output:

Next page

Enter a string : prathamesh

a occurs 2 times in the string

e occurs 1 times in the string

h occurs 2 times in the string

m occurs 1 times in the string

P occurs 1 times in the string

r occurs 1 times in the string

s occurs 1 times in the string

t occurs 1 times in the string

## Experiment No: 8

**Aim:**

**Study of structures.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	<i>First Year and D</i>	<i>15/07/2021</i>	<i>22/07/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

**Program list:**

- 1. Write a program to read and print an employee's detail using structure.**

PAGE NO	01
DATE	/ /

Program ProblemsProgram 1problem:

write a program to read and print an employee's detail using structure.

Program:

#include&lt;stdio.h&gt;

Struct employee

{

int e;

char name [20];

char designation [20];

char dept [20];

int sal;

};

int main ()

{

Struct employee a;

printf ("Enter Employee Details:\n");

printf ("---\n");

printf ("Enter Employee - Id : ");

scanf ("%d", &amp;a.e);

printf ("Enter Name : ");

scanf ("%s", a.name);

printf ("Enter designation : ");

scanf ("%s", a.designation);

printf ("Enter Department : ");

scanf ("%s", a.dept);

printf ("Enter salary : ");

scanf ("%d", &amp;a.sal);

printf ("---");

```

printf ("Enter Employee Details : \n");
printf ("Employee-ID : %d\n", a.e);
printf ("Name : %s\n", a.name);
printf ("Designation : %s\n", a.designation);
printf ("Department : %s\n", a.dept);
printf ("Salary : %d\n", a.salary);
return 0;
}

```

Output:

Enters Employee Details:

---  
Enters Employee-ID : 3322

Enters Name : Samuel

Enters Designation : Manager

Enters Department : xyz Agency

Enters Salary : 100000  
---

Employee Details:

---  
Employee-ID : 3322

Name : Samuel

Designation : Manager

Department : xyz Agency

Salary : 100000

## Experiment No: 9

Aim:

**Study of pointers.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> PRATHAMESH SHIVAJI CHIKANKAR	<i>First Year and D</i>	23/06/2021	22/07/2021
<b>Roll no:</b> AIMLD08			

**Program list:**

- 1. Write a program to swap two integer numbers using pointers.**

Program 1:Problem:

write a program to swap two integers numbers using pointers.

Program

```
#include <stdio.h>
int main()
{
    int x, y, *a, *b, temp;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before swapping \nx = %d\ny = %d\n", x, y);
    a = &x;
    b = &y;
    temp = *b;
    *b = *a;
    *a = temp;
    printf("After swapping \nx = %d\ny = %d\n", x, y);
    return 0;
}
```

Output:

Enter the value of x and y

77 97

Before swapping

x = 77

y = 97

After swapping

x = 97

y = 77

## Experiment No: 10

**Aim:**

**Study of sum of series and pattern programs.**

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> PRATHAMESH SHIVAJI CHIKANKAR	<i>First Year and D</i>	23/06/2021	22/07/2021
<b>Roll no:</b> AIMLD08			

**Program list:**

- 1. WAP to calculate sum of series of  $1/1! + 2/2! + 3/3! + \dots + N/N!$**
- 2. WAP to display following pattern:**

1			
2	3		
4	5	6	
7	8	9	10

Program 1:problem:

WAP to calculate sum of series of  $1/1! + 2/2! + 3/3! + \dots + n/n!$

Program:

```
#include <stdio.h>
#include <conio.h>
double sumseries(double); main()
{
    int i, n;
    double number, sum;
    printf("Enter the number : ");
    scanf("%lf", &number);
    sum = sumseries(number);
    printf("The sum of the above series = %lf", sum);
    getch();
}

double sumseries(double m)
{
    double sum2 = 0, f = 1, i;
    for (i = 1; i <= m; i++)
    {
        f = f * i;
        sum2 = sum2 + (i / f);
    }
    if (i == m)
        printf("%.2f / %.2f", i, f, sum2);
    else
        printf("%.2f / %.2f + \n", i, f);
    return sum2;
}
```

output:

Ents the number : 5

1.00 / 1.00 +

2.00 / 2.00 +

3.00 / 6.00 +

4.00 / 24.00 +

5.00 / 120.00 = 2.708333

Sum of the above series = 2.708333

program 2

problem:

WAP to display following patterns:

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

program:

```
#include<stdio.h>
#include<conio.h>
main()
{
    int i,j,k=1;
    for(i=1; i<=5; i++)
    {
        for(j=1; j<i; j++)
            printf("%d", k);
        k++;
        printf("\n");
    }
    getch();
}
```

PAGE No.	03
DATE	/ /

Output:

1  
2 3  
4 5 6  
7 8 9 10

The output of above

the output contains

double sum and float round

float sum, float sum

double number, sum of all numbers

float f (float numbers)

float f (float numbers)

sum of numbers (number)

float f (sum of the above series = float (sum))

which is 25.0

float sum = 0.0

double sum = 0.0, f = 0.0

double sum = 0.0, f = 0.0

float f (float numbers)

## Assignment No: 1

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	<i>First Year and D</i>		<i>23/07/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

- 1. Enlist all the data types in C language along with memory requirements, format specifiers and range.**
- 2. Distinguish between**
  - i) while and do while statement**
  - ii) continue and break statement**
- 3. explain storage class.**
- 4. What is computer programming. What do you mean by structured programming? Develop an algorithm and flowchart to find reverse of a number.**
- 5. Enlist bitwise operators in C language. Explain any two with example.**

1. Enlist all the data types in C language along with memory requirements, format specifiers and range.

### Data types

These are used to declare variables or functions of different types. This determines type of variable how much space it occupies in storage.

C has five basic data types and they are as follows:

- Character - Keyword used is `char`
- Integer - Keyword used is `int`
- floating point - Keyword used is `float`
- double precision floating point - Keyword used is `double`
- Valueless - Keyword used is `void`.

Data type	Range	Bytes	Format
Signed char	-128 to +127	1	%c
unsigned char	0 to 255	1	%c
short int	-32768 to +32767	2	%d
short Unsigned	0 to 65535	2	%u
Signed int	-2147483648 to +2147483647	4	%d
Unsigned int	0 to 4294967295	4	%u
float	-3.4e38 to +3.4e38	4	%f
double	-1.7e308 to +1.7e308	8	%lf
long double	-1.7e4932 to +1.7e4932	10	%Lf

Note: The sizes and ranges of int, short and long are compiler dependent.

Sizes in bytes are for 32-bit compilers.

## AIML008

2. Distinguish between  
 i) while and do-while statement  
 ii) continue and break statement

i) while	do-while
1. Condition is checked first then statement(s) is executed.	statement(s) is executed atleast once, thereafter condition is checked again and if condition is true then statement(s) is executed again.
2. It might occurs statement(s) is executed zero times, if condition is false.	At least once the statement(s) is executed.
3. No semicolon at the end of while. while (condition)	Semicolon at the end of while (condition);
4. If there is single statement, brackets aren't required.	Brackets are always required
5. while loop is entry controlled loop.	do-while is loop of exit controlled loop.

ii) Break	Continue
1. The break statement is used to exist from the loop constructs.	The continue statement is not used to exist from the loop constructs.
2. The break statement is usually used with the switch statement, and it can also use it within the while loop, do-while loop or the for-loop.	The continue statement is not used with the switch statement, but it can be used within the while loop, do-while loop, or for-loop.
3. Syntax:  break;	Syntax:  continue;

AIML08

3.1 Explain storage class!

**Ans:-** Storage classes are used to describe the features of a variable / function. These features basically include the scope, visibility and life-time which help us to have the existence of a particular variable during the run-time of a program.

### // Storage classes in C //

storage Specifier	Storage	initial value	scope	life
auto	Stack	Garbage	within block	End of block
extern	Data Segment	zero	global multiple files	Till end of program
static	data segment	zero	within block	Till end of program
register	CPU register	Garbage	within block	End of block

4. what is computer programming. what do you mean by structured programming? Develop an algorithm and flowchart to find reverse of a number.

Computer programming is the process of designing and building an executable computer program to accomplish a specific computing result to or to perform a specific task.

Structured programming is a programming paradigm

AIML08

aimed at improving the clarity, equality and development time of a computer program by making extensive use of the structured controlled flow constructs of selection (if/then/else) and repetition (while & for), block structures and subroutines in contrast to using simple tests and jumps such as the `goto` statement.

## Reverse of a Number

### Algorithm

Step 0: Start

Step 1: Initialize  $rev = 0$

Step 2: Input  $n$ ,  $sem$  and integers type value.

Step 3: Print Enter the an integers

Step 4: Input the value in  $n$

Step 5: Continue Step 6 - Step 8 till  $n = 0$

Step 6:  $sem = n \% 10$

Step 7:  $rev = rev * 10 + sem$ .

Step 8:  $n = n / 10$

Step 9: Print "Reversed number = " & display the value of  $reverse$

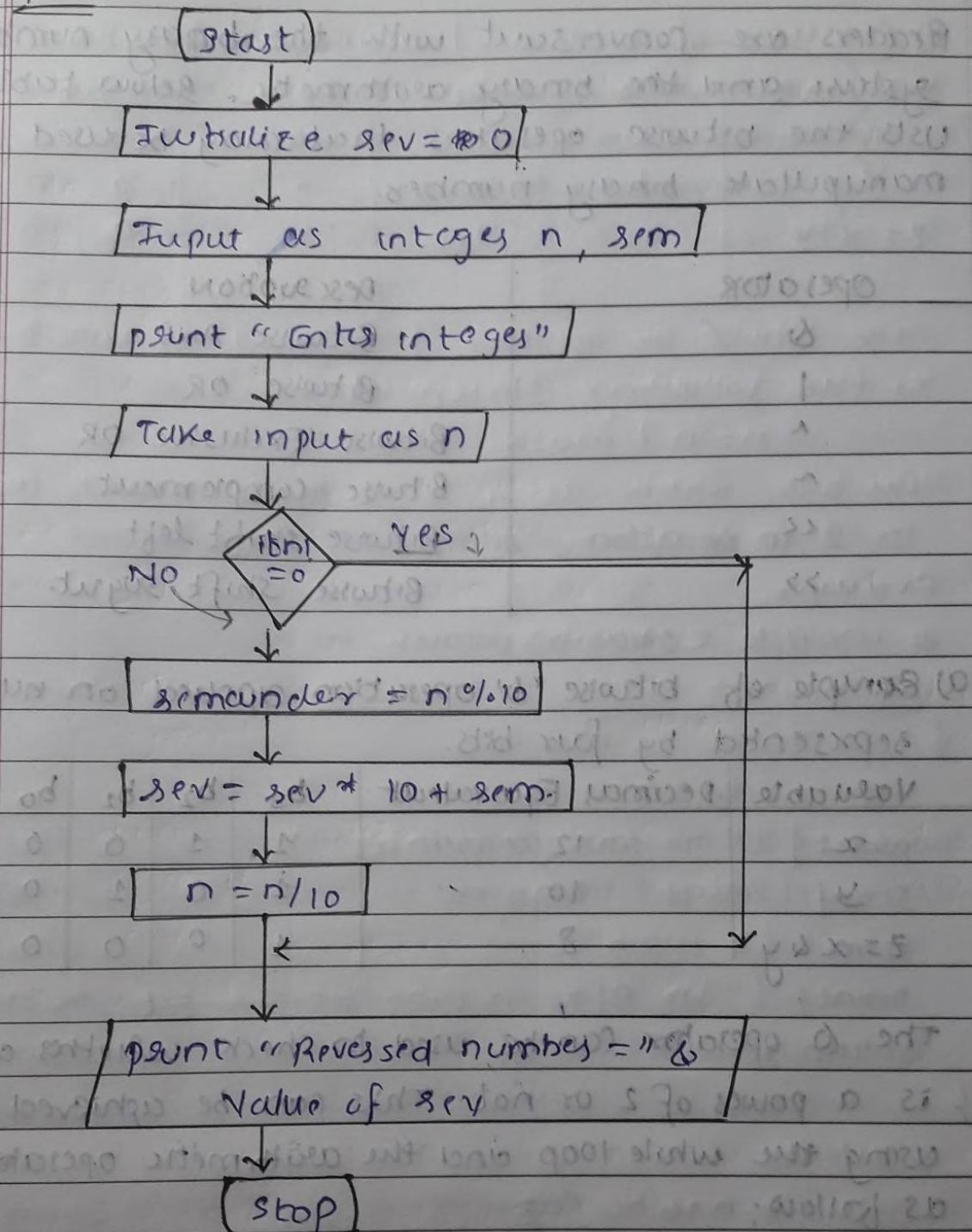
Step 10: Stop.

### Flowchart

Next page

flowchart!

AIMLD08



5. Explain bitwise operators in C language.  
Explain any two with example.

**Ans:-** Since a computer understands only machine language, data is represented as binary numbers that are nothing but various combinations of 0's & 1's.

## AIML D08

Readers are conversant with the binary number system and the binary arithmetic. Below table lists the bitwise operators that may be used to manipulate binary numbers.

Operator	Description
&	Bitwise AND
	Bitwise OR
^	Bitwise Exclusive OR
~	Bitwise complement
<<	Bitwise shift left
>>	Bitwise shift right

a) Example of bitwise '&' operation applied on numbers represented by four bits.

Variable	Decimal Equivalent	$b_3$	$b_2$	$b_1$	$b_0$
x	12	1	1	0	0
y	10	0	1	0	1
$z = x \& y$	8	1	0	0	0

The & operator can be used to check whether a number is a power of 2 or not. This can be achieved by using the while loop and the arithmetic operator % as follow:

Example:

```
#include <stdio.h>
int main()
{
    int n,r;
    printf("Enter the number:");
    scanf("%d", &n);
    r=n%2;
    if(r==0)
        printf("The number is a power of 2");
    else
        printf("The number is not a power of 2");
}
```

AIM LD08

```

scanf ("%d", &n); // n is assigned the value obtained from the user
while (n > 1) {
    r = n % 2; // r is assigned the value obtained as remainder
    if (r == 0)
        n = n / 2;
    else
        break;
}
if (r != 0)
    printf ("\\n The number is not power of 2");
else
    printf ("\\n The number is power of 2");
return 0;
}

```

Using bitwise AND, the program in above example can be rewritten without using the loop or the arithmetic operators.

```

#include <stdio.h>
int main()
{
    int n;
    printf ("\\nEnter the number:\\n");
    scanf ("%d", &n);
    if ((n & (n - 1)) == 0) // bitwise operation
        printf ("\\n The number is power of 2");
    else
        printf ("\\n The number is not power of 2");
    return 0;
}

```



## Assignment No: 2

Name and roll no of student	Class and Division	Date of performance	Date of submission
<b>Name:</b> <i>Prathamesh Shivaji Chikankar</i>	<i>First Year and D</i>		<i>23/07/2021</i>
<b>Roll no:</b> <i>AIMLD08</i>			

- 1. What is an array? What does an array name signify? Can array index be negative? Write a program to arrange the number stored in an array in such a way that the array will have the odd numbers followed by even numbers.**
- 2. Distinguish between structure and union. Explain concept of nested structure. Declare a structure to enter employee Information like name, id, salary, date of joining. Use nested structure to get the address of an employee. Write a program to read 10 records and display them.**
- 3. What is string? Explain the use of gets ()? Write a c program that will read a word and rewrite it in alphabetical order. For ex. If the word is "matrix" the program should print "aimrtx".**
- 4. What is pointer. Explain how the pointer variable is declared and initialized.**
- 5. Explain string functions with example.**

1. what is an array? what does an array name signify? can array index be negative? write a program to arrange the numbers sorted in an array in such a way that the array will have the odd numbers followed by even numbers.

Ans:- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.

Array name signifies that the pointer of an array is the pointer of its first element. The name of an array is the pointer to the first element of the array. If 'p' is a pointer to array 'age', means that p(or age) points to age[0].

Yes, array index can be negative as that allows negative index in multi arrays since it is just a contiguous chunk of memory and index lookups is just pointer arithmetic on it.

program to sort sort even and odd elements

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a[10000], b[10000], i, j, K, tmp, c=0;
    printf("Enter size of the array : ");
    scanf("%d", &n);
    printf("Enter elements in array : ");
    for (i=0; i<n; i++)
    {
        Scanf("%d", &a[i]);
    }
    for (i=0; i<n; i++)
    {
        for (j=i+1; j<n; j++)
        {
            if (a[i] > a[j])
            {
                tmp = a[i];
                a[i] = a[j];
                a[j] = tmp;
            }
        }
    }
    for (i=0; i<n; i++)
    {
        if (i % 2 == 0)
        {
            b[c] = a[i];
            c++;
        }
    }
    for (i=0; i<c; i++)
    {
        printf("%d ", b[i]);
    }
}
```

AIMLD08

```
if (a[i] % 2 == 1)
```

```
c++; // increment count of odd numbers
```

```
for (i = 0; i < n - 1; i++)
```

```
for (j = 0; j < n - i - 1; j++)
```

```
{ if (a[j] > a[j + 1])
```

```
temp = a[j];
```

```
a[j] = a[j + 1];
```

```
a[j + 1] = temp;
```

```
}
```

```
K = 0;
```

```
j = n - c;
```

```
for (i = 0; i < n / 2; i++)
```

```
{
```

```
if (a[i] % 2 == 0)
```

```
{
```

```
if (K < n - c)
```

```
b[K++] = a[i];
```

```
}
```

```
else
```

```
{
```

```
if (j < n)
```

```
b[j++] = a[i];
```

```
}
```

```
}
```

800 MINTER

printf ("In array after sorting even and odd  
elements separately : In ");

```
for (i=0; i<n; i++)
```

{

```
a[i] = b[i];
```

```
printf ("%d", a[i]);
```

}

}

2. Distinguish between structure and union. Explain concept of nested structure. Define structure to enter employee information like name, id, salary date of joining. Use nested structure to get the address of an employee. Write a program to read 10 records and display them.

Ans:-

	Structure	Union
Keyword	Keyword struct is used to define a structure.	Keyword union to define union.
Size	Size of structure is greater than or equal to sum of size of its members.	Size of union is equal to the size of largest members.
Memory	Each members within a structure is assigned unique storage area of location.	memory allocated is shaped by individual members of union.
Accessing members	Individual members can access at a time.	Only one members can be accessed at a time.
Initialization of members	Several members of a structure can initialize at once.	only first members of a union can be initialized.

Nested Structure (also known as "Struct Inside")  
when a structure contains another structure, it is called nested structure. For example, we have two structures named Address and Employee. To make Address nested to Employee, we have to define Address structure before and outside Employee structure and create an object of address structure inside Employee structure.

Program:

```
#include <stdio.h>
struct Employee
{
    int Id;
    char Name[25];
    int Age;
    long Salary;
};

main()
{
    struct Employee Emp[10];
    for(i=0; i<10; i++)
        printf("Enter details of %d employee", i+1);
    printf("Enter Employee Id: ");
    scanf("%d", &Emp[i].Id);
    printf("Enter Employee Name: ");
    scanf("%s", &Emp[i].Name);
```

BSCS101A

AIM1008

```
printf("It enters Employee Age : ");
scanf("%d", &Emp[i].Age);
```

```
printf("It enters Employee salary : ");
scanf("%d", &Emp[i].Salary);
}
```

```
printf("Details of Employees");
for(i=0; i<10; i++)
printf("\n%d\t%s\t%d\t%d", Emp[i].Id,
      Emp[i].Name, Emp[i].Age, Emp[i].Salary);
}
```

3. what is string? Explain the use of gets ()?  
 write a C program that will read a word and  
 reverse it in alphabetical order. For ex. If the  
 word is "matrix" the program should print "xitmra".

Ans:- strings are actually one-dimensional array of  
 element characters terminated by a null character  
 (\0). Thus a null-terminated string contains the  
 characters that comprise the string followed by a null.  
 The gets() function enables the user to enter some  
 characters followed by the enter key. All the characters  
 entered by the user gets stored in a character array. The  
 null character is added to the array to make it a string.  
 It returns string entered by the user.

Program:

Next page.

```

#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <string.h>
main()
{
    char x[50], temp;
    int i, j, n;
    printf("Enter a string!");
    gets(x);
    n = strlen(x);
    for (i = 0; i < n - 1; i++)
        for (j = 0; j < n - 1 - i; j++)
            if (tolower(x[j]) > tolower(x[j + 1]))
                {
                    temp = x[j];
                    x[j] = x[j + 1];
                    x[j + 1] = temp;
                }
    printf("Given string in alphabetical order is: %s", x);
}

```

Q. What is pointer? Explain how the pointer variable is declared and initialized.

Ans:- A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. The general form of pointer declaration:

800 MCA

AIML D08

is: type \* var-name;

**Declaration:** The general syntax of pointer declaration is,

datatype \* pointer-name;

The data type of the pointer and the variable to which the pointer variable is pointing must be the same.

**Initialization:**

pointer initialization is the process of assigning address of a variable to a pointer variable. It contains the address of a variable of the same data type. In C language address operator & is used to determine the address of a variable. The & (immediately preceding a variable name) returns the address of the variable associated with it.

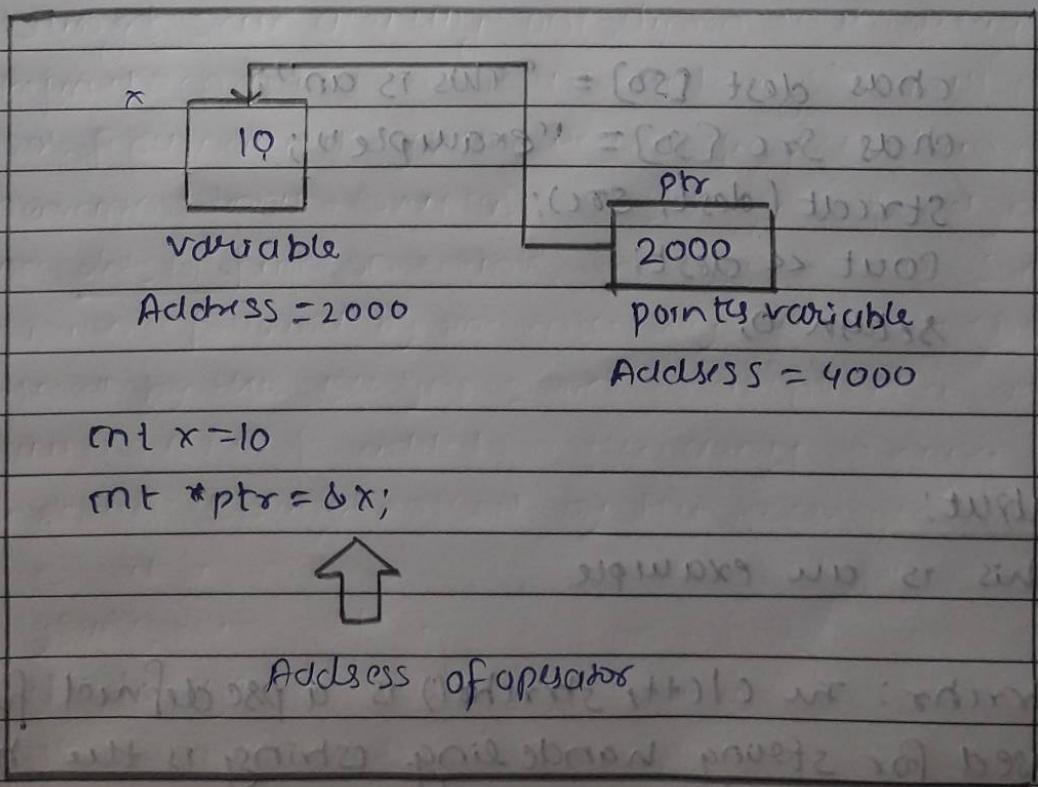
```
int a=10;
```

```
int *ptr;
```

```
ptr = &a;
```

// pointer declaration

// pointer initialization



5. Explain string functions with example.

Ans:- Most commonly used string functions are:

- **strcat:** The `strcat()` function will append a copy of the source string to the end of destination string. The `strcat()` function takes two arguments:

1) dest

2) src

It will append copy of the source string in the destination string. The terminating character at the end of dest is replaced by the first character of src. Return value: The `strcat()` function returns dest, the pointer to the destination string.

Program:

```
#include <cstring>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
char dest [50] = "This is an";
```

```
char src [50] = "example";
```

```
strcat (dest, src);
```

```
cout << dest;
```

```
return 0;
```

```
}
```

Output:

This is an example

- **strchr:** In C/C++, `strchr()` is a pre-defined function used for string handling. `cstring` is the header

file required for string functions.

Syntax:

`char * strchr (const char *str, int c)`

Example: program

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

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    char st[] = "Geeks for Geeks";
```

```
    char ch = 'e';
```

```
    char * val;
```

```
    val = strchr (st, ch)
```

```
    printf ("String after last %c is : %s \n", ch, val);
```

```
    char ch2 = 'm';
```

```
    val = strchr (st, ch2);
```

```
    printf ("String after last %c is : %s ", ch2, val);
```

```
    return (0);
```

```
}
```

Output:

String after last e is : eks

String after last m is : (null)

- strcmp
- strcpy
- strlen
- strcat
- strncmp
- strncpy
- strchr