



G. H. Raisoni
College of Engineering & Management

Wagholi, Pune

Certificate

This is to certify that

Mr./Ms. Pratik Rajesh Jade

studying in Semester/Year First year

Branch Artificial Intelligence

Name of the Dept. F. Y. B. tech

Section A

has satisfactorily completed the practical work of

subject Computer Programming

during the academic year 2020-2021

Signature of Subject Teacher

Signature of Head of Department



INDEX PAGE



RAISONI GROUP
— A vision beyond —

LAB EXERCISE #1

Objective(s):

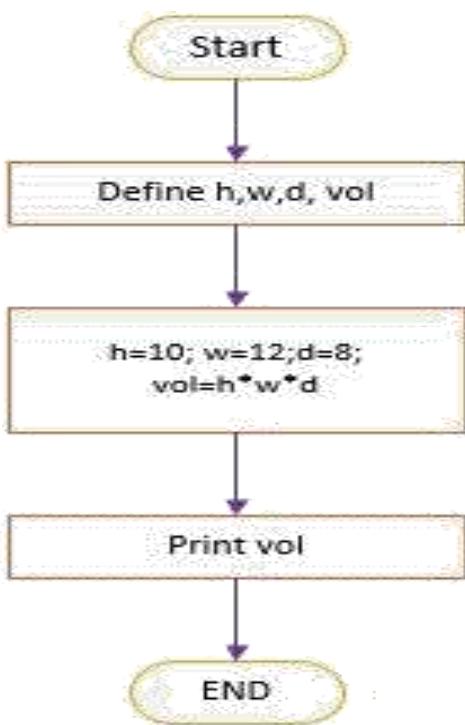
To be familiar with syntax and structure of C- programming.
To learn problem solving techniques using C

Program: Write a Program to calculate and display the volume of a CUBE having its height (h=10cm), width (w=12cm) and depth (8cm).

Algorithm:

1. Start
2. Define variables: h(int), w(int), d(int), vol(int)
3. Assign value to variables: h = 10, w=12, d=8
4. Calculate the volume as: vol = h*w*d
5. Display the volume (vol)
6. Stop

Flowchart:



Code: (*Use comments wherever applicable*)

```
//Following code is written and compiled in GCC

#include<stdio.h>
void main()
{
//start the program
int h,w,d,vol; //variables declaration
h=10;w=12;d=8; //assign value to variables
vol=h*w*d;      //calculation using mathematical formula
printf("The Volume of the cube is: %d",vol); //display the
volume
getch();
//end the main program
}
```

Output :

The Volume of the cube is: 960

LAB EXERCISE #2

Objective(s):

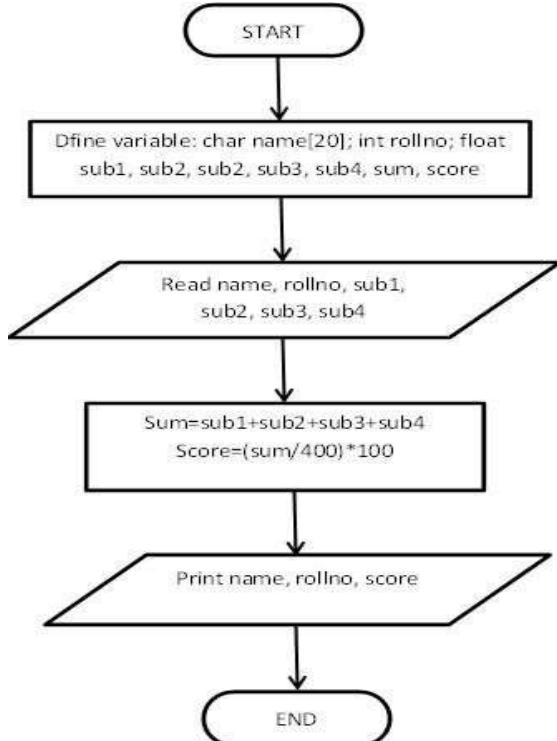
To be familiar with different data types, Operators and Expressions in C.

Program: Write a program to take input of name, rollno and marks obtained by a student in 4 subjects of 100 marks each and display the name, rollno with percentage score secured.

Algorithm:

1. Start
2. Define variables: name, rollno, sub1, sub2, sub3, sub4, sum, score
3. Take input from keyboard for all the input variables
4. Calculate the sum of marks of 4 subjects and also calculate the percentage score as:
$$\text{sum} = \text{sub1} + \text{sub2} + \text{sub3} + \text{sub4};$$
$$\text{score} = (\text{sum}/400) * 100$$
5. Display the name, roll number and percentage score.
6. Stop

Flowchart:



Code: (*Use comments wherever applicable*)

```
//Following code is written and compiled in TURBO C++  
  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
char name[20];  
int rollno;  
float sub1, sub2, sub3, sub4, , sum, score;  
printf("Enter name of student: ");  
scanf("%s",&name[]);  
printf ("\n Enter Roll Number: ");  
scanf("%d", &rollno);  
printf (" \n Enter Marks in 4 Subjects:\n");  
scanf("%f%f%f%f", &sub1, &sub2, &sub3, &sub4);  
sum=sub1+sub2+sub3+sub4;  
score = (sum/500)*100;  
printf("\n Name of student: %s", name[]);  
printf("\n Roll Number: %d", rollno);  
printf ("\nPercentage score secured: %2.2f%c", score, '%');  
getch();  
}
```

Output:

Enter name of student: Ajit Singh

Roll Number: 25

Enter Marks in 4 Subjects:

50

75

85

62

Name of student: Ajit Singh

Roll Number: 25

Percentage score secured: 68.00%

LAB EXERCISE #3

Objective(s):

To understand the programming knowledge using Decision Statements (if, if-else, if-else-if ladder, switch and GOTO)

Program: Write a program to print whether a given number is even or odd.

Code: (*Use comments wherever applicable*)

```
//Following code is written and compiled in TURBO C++  
  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    int num;  
    printf("Enter the number: ");  
    scanf("%d",&num);  
    if(num%2==0)  
        printf("\n %d is even", num);  
    else  
        printf("\n %d is odd", num);  
    getch();  
}
```

Output:

Enter the number: 6

6 is even

Program: Write a program to find whether a character is consonant or vowel using switch statement.

```
#include <stdio.h>
void main()
{
char ch;
printf("Enter any alphabet:"); //input alphabet from user
scanf("%c", &ch);
switch(ch)
{
    case „a“:
    case „A“:
        printf("Vowel");
        break;
    case „e“:
    case „E“:
        printf("Vowel");
        break;
    case „i“:
    case „I“:
        printf("Vowel");
        break;
    case „o“:
    case „O“:
        printf("Vowel");
        break;
    case „u“:
    case „U“:
        printf("Vowel");
        break;
    default:
        printf("Consonant");
}
}
```

LAB EXERCISE #4

Objective(s):

To understand the programming using Loop & nested loop Statements (for, while, do-while)

Program: Write a program to print positive integers from 1 to 10.

Code:

```
//Following code is written and compiled in TURBO C++  
//Using FOR LOOP  
  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int i;  
for(i=1; i<=10;i++)  
    printf("%d \n", i);  
getch();  
}  
  
//Using WHILE LOOP  
  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int i=1;  
while(i<=10)  
{  
    printf("%d \n", i);  
    i++;  
}  
getch();  
}
```

```
/Using DO-WHILE LOOP

#include<stdio.h>
#include<conio.h>
void main()
{
int i=1;
do
{
    printf("%d \n", i);
    i++;
}
while(i<=10);
getch();
}
```

Output:

1
2
3
4
5
6
7
8
9
10

Program: Write a program to display the following pattern.

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Code:

```
#include  
#include  
void main()  
{  
int i,j;  
for(i=1; i<=5;i++)  
{  
for(j=1;j<=i;j++)  
{  
printf("*");  
}  
printf("\n");  
}  
getch();  
}
```

LAB EXERCISE #5

Objective(s):

To understand programming using different dimensions of Array.

Program: Write a program to insert 5 elements into an array and print the elements of the array.

Code: (*Use comments wherever applicable*)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, arr[5];
    printf("Enter the elements into the array:");
    for(i=0; i<=4;i++)
        scanf("%d", &arr[i]);
    printf("The elements of the array are:");
    for(i=0; i<=4;i++)
        printf("%d \t", arr[i]);
    getch();
}
```

LAB EXERCISE #6

Objective(s):

To understand function programming, its types and function-call.

Program: Write a program to calculate factorial of a number using recursion.

Code:

```
#include<stdio.h>
long factorial(int); //Function declaration
int main()
{
    int num;
    long fact;
    printf("Enter a number to find factorial: \n");
    scanf("%d", &num);
    if(num<0)
        printf("Factorial of negative no. is not defined. \n");
    else
    {
        fact = factorial(num);
        printf("%d!=%d \n", num, fact);
    }
    return 0;
}
//Function definition
long factorial(int num)
{
    if(num==0)
        return 1;
    else
        return (num*factorial(num-1));
}
```

LAB EXERCISE #7

Objective(s):

To understand programming with Pointer, String and Function call by reference.

Program: Write a program to find biggest among three numbers using pointer.

Code:

```
#include<stdio.h>
#include<conio.h>
int main()
{
int a,b,c;
int*ptra=&a,*ptrb=&b,*ptrc=&c;
printf("enter three values");
scanf("%d%d%d",ptra,ptrb,ptrc);
printf("a=%d\n b=%d\n c=%d\n",*ptra,*ptrb,*ptrc);
if((*ptra>*ptrb && *ptra>*ptrc))
    printf("biggest number=%d",*ptra);
else if((*ptrb>*ptra && *ptrb>*ptrc))
    printf("biggest number =%d",*ptrb);
else
    printf("biggest number=%d",*ptrc);
getch();
return 0;
}
```

LAB EXERCISE #8

Objective(s):

To understand programming with Structure.

Program 1: Write a C program to create, declare and initialize structure.

Code:

```
#include <stdio.h>
/*structure declaration*/
struct employee{
    char name[30];
    int empId;
    float salary;
};

int main()
{
    /*declare and initialization of structure variable*/
    struct employee emp={"Anil",201,80000.00};

    printf("\n Name: %s      ,emp.name);
    printf("\n Id: %d      ,emp.empId);
    printf("\n Salary: %f\n",emp.salary);
    return 0;
}
```

Program 2: Write a program to store information of 5 students in structure and display it.

Code:

```
#include<stdio.h>
struct student
{
    char name[30];
    int roll;
    float marks;
}s[5];
int main( )
{
    int i;
    printf("Information of students:");
}
```

```

for (i=0; i<5; ++i)
{
    s[i].roll =i+1;
    printf("\n Roll number %d, \n", s[i].roll);
    printf("Enter name:");
    scanf("%s", s[i].name);
    printf("Enter marks:");
    scanf("%f", &s[i].marks);
}
printf("\n Displaying Information:\n");
for(i=0;i<10;++i)
{
    printf("\n Roll number:%d \n", i+1);
    printf("Name:");
    puts(s[i].name);
    printf("\n Marks:%.1f", s[i].marks);
}
return 0;
}

```

Program 3: Write a program to declare, initialize an UNION.

Code:

```

#include <stdio.h>
// union declaration
union pack{
char a;
int b;
double c;
};
int main()
{
    pack p; //union object/variable declaration
    printf("\nOccupied size by union pack:
%d", sizeof(pack));
    // assign value to each member one by one other it
will replace last value
    p.a='A';
    printf("\nValue of a:%c",p.a);
    p.b=10;
    printf("\nValue of b:%d",p.b);
    p.c=12345.6790;
    printf("\nValue of c:%f",p.c);
}

```

```
// see, what will happen? if u will assign values
together
p.a='A';
p.b=10;
p.c=12345.6790;
// here the last value of p.c will be accessed by all
members
printf("\nValue of a:%c, b:%d, c:%f",p.a,p.b,p.c);
return 0;
}
```

LAB EXERCISE #9

Objective(s):

To understand data files and file handling in C.

Program 1: Write a program to create a file called emp.rec and store information about a person, in terms of his name, age and salary.

Code:

```
#include <stdio.h>
void main()
{
    FILE *fptr;
    char name[20];
    int age;
    float salary;
    /* open for writing */
    fptr = fopen("emp.rec", "w");
    if (fptr == NULL)
    {
        printf("File does not exists \n");
        return;
    }
    printf("Enter the name \n");
    scanf("%s", name);
    fprintf(fptr, "Name      = %s\n", name);
    printf("Enter the age\n");
    scanf("%d", &age);
    fprintf(fptr, "Age      = %d\n", age);
    printf("Enter the salary\n");
    scanf("%f", &salary);
    fprintf(fptr, "Salary   = %.2f\n", salary);
    fclose(fptr);
}
```

Program 2: Write a program to illustrate how a file stored on the disk is read.

Code:

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    FILE *fptr;
    char filename[15];
    char ch;
    printf("Enter the filename to be opened \n");
    scanf("%s", filename);
    /* open the file for reading */
    fptr = fopen(filename, "r");
    if (fptr == NULL)
    {
        printf("Cannot open file \n");
        exit(0);
    }
    ch = fgetc(fptr);
    while (ch != EOF)
    {
        printf ("%c", ch);
        ch = fgetc(fptr);
    }
    fclose(fptr);
}
```

Name of Student: Pratik Jade

Roll no: A72

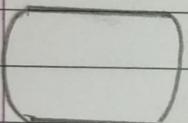
Experiment no. 01

Title: Implement Syntax of C with algorithm & flowchart

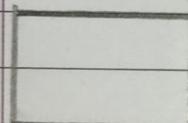
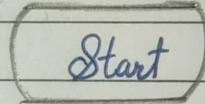
1. Write the Symbol and specification of each symbol used in flowchart.



Symbol	Type of operation/C code example	Explain example of symbol usage.
--------	----------------------------------	----------------------------------



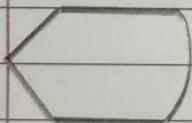
Terminal Activity - Start, Stop or end



Assignment of a value to a variable, either directly or as the result of a calculation.

$$I = I + 1;$$

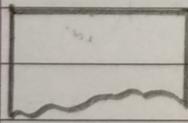
ADD 1 to I
or
 $I = I + 1$



Softcopy - Screen output to a video display.

point F ("The answer is : %d, A);

The answer is : A



Hardcopy - document output from a printer.

The answer is : A



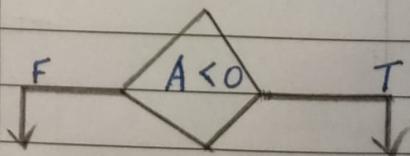
Decision based on a condition.

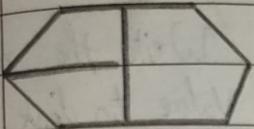
if ($A < 0$)

{

Statement;

else
statement;

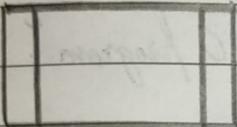
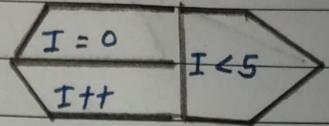




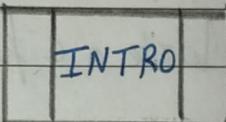
To repeat a Statement/s a known number of times.

for ($I = 0$; $I < 5$; $I++$)

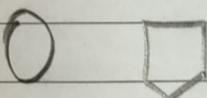
 Statement;
 }



Sub-routine (function) used to indicate a process, which is defined elsewhere.

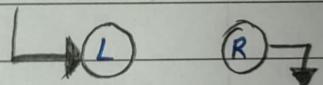


Intro(); /* Call Intro */



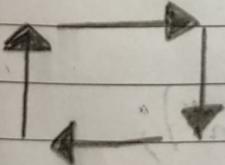
Connectors:

On-page (left) & off-page (right).

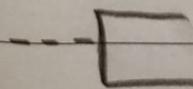
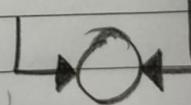


Used to either:

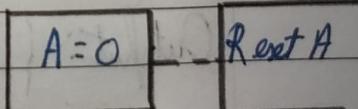
1. Continue a flowchart at a diff place either on or off the same piece of page.
2. Close a Selection branch or loop.

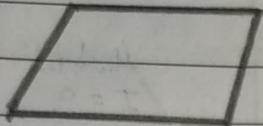


Flow of Control Arrows indicating the sequence of steps ("flow of control").



Annotation for placing comments in logic
 $A = 0$;
/* Reset A */





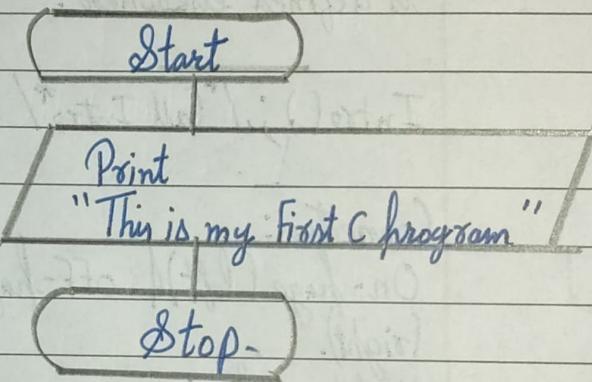
General Input / Output of Data

```
fprintf (filename, data);
printf ("Message");
scanf ("%d, & I);
```

Write the next
Value to disk

2 Write a c program to display "This is my first C program".

Flow Chart:-



Program:-

```
#include <stdio.h>
int main() {
    printf("Hello,
```

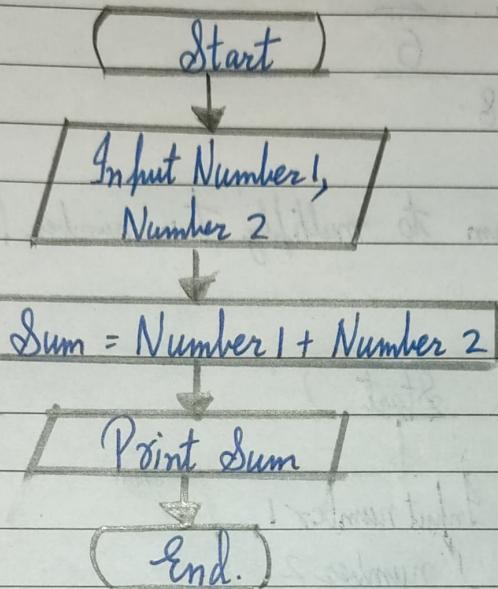
```
#include <stdio.h>
int main() {
    printf("This is my First C program");
    return 0;
}
```

Output:-

This is my First C program

3 Write a c program to add two number (2 and 6) & display its sum.

→ Flow chart



Program:

```

#include <stdio.h>
int main()
{
    int number1, number2, sum;

    printf("Enter number 1: ");
    scanf("%d", &number1);

    printf("Enter number 2: ");
    scanf("%d", &number2);

    sum = number1 + number2;

    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}
  
```

Output :

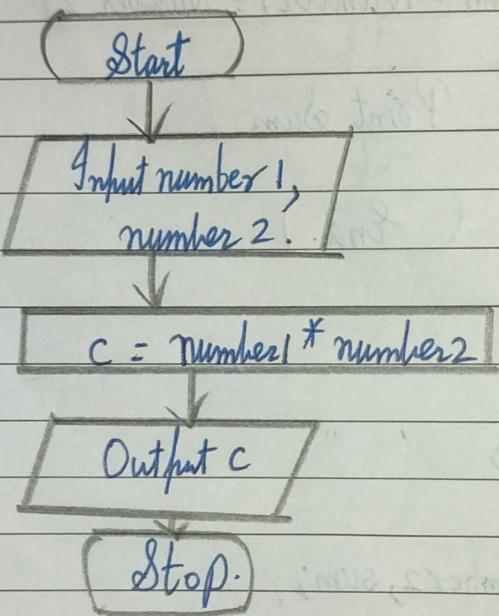
Enter number 1 : 2

Enter number 2 : 6

$$2 + 6 = 8$$

- 4 Write a C program to multiply two numbers (4 & 5) & display its product

→ Flowchart:



Program:

```

#include <stdio.h>
int main(){
    int number1, number2, sum;
    #del("Enter number 1:");
    printf("Enter number 1:");
    scanf("%d", &number1);
    printf("Enter number 2:");
    scanf("%d", &number2);
    sum = number1 * number2;
    printf("Sum = %d", sum);
}
  
```

Sum = number1 * number2;

```
pointF ("%.d * %.d = %.d", number1, number2, sum);
return 0;
}
```

Output:

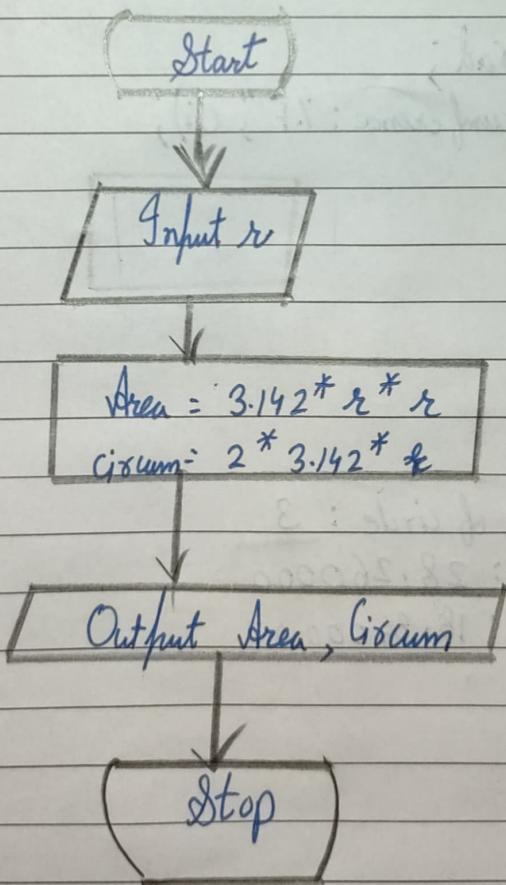
Enter number1 : 4

Enter number2 : 5

$$4 * 5 = 20$$

5 Write a c program to calculate area & circumference of a circle.

→ Flowchart:



Program:

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

```
int main() {
```

```
    int rad;
```

```
    float PI = 3.14, area, ci;
```

```
    printf ("\nEnter radius of circle: ");
```

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

```
    area = PI * rad * rad;
```

```
    printf ("\nArea of circle: %.2f", area);
```

```
    ci = 2 * PI * rad;
```

```
    printf ("\nCircumference: %.2f", ci);
```

```
    return (0);
```

```
}
```

Output:

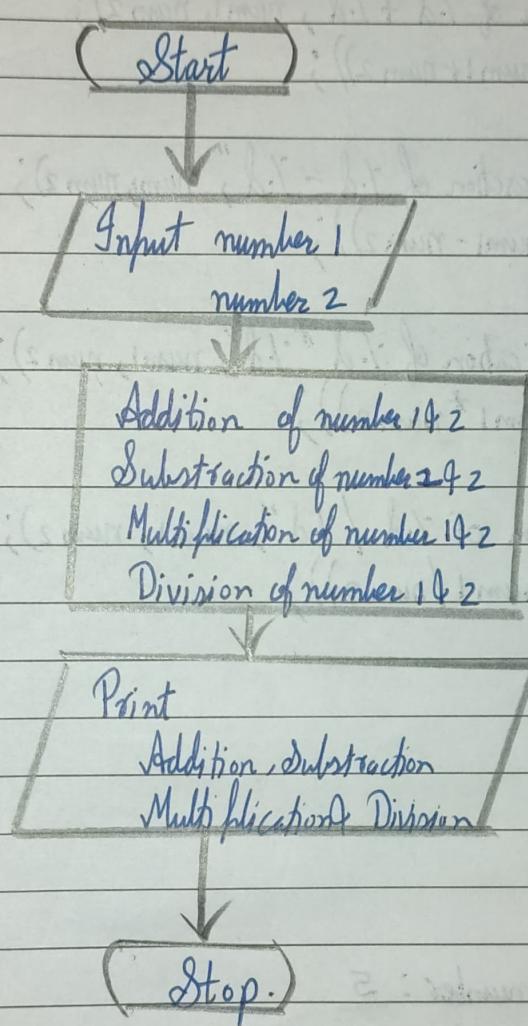
Enter radius of circle : 3

Area of circle : 28.260000

Circumference : 18.840000

6 Write a C program to perform addition, subtraction, division & multiplication of two numbers.

→ Flowchart



Program.

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int num1, num2;
    printf("Enter the first number: ");
    scanf("%d", &num1);
  
```

```
printf("Enter the second number: ");
scanf("%d", &num2);
```

```
printf("\nAddition of %.d + %.d", num1, num2);
printf(" = %.d", (num1+num2));
```

```
printf("\nSubtraction of %.d - %.d", num1, num2);
printf(" = %.d", (num1-num2));
```

```
printf("\nMultiplication of %.d * %.d", num1, num2);
printf(" = %.d", (num1 * num2));
```

```
printf("\nDivision of %.d / %.d", num1, num2);
printf(" = %.d", (num1 / num2));
```

```
return 0;
}
```

Output:-

Enter the First number: 5

Enter the Second number: 4

Addition of $5 + 4 = 9$

Subtraction of $5 - 4 = 1$

Multiplication of $5 \times 4 = 20$

Division of $5 / 4 = 1$

Name of Student - Pratik Rajesh Jade

Roll no - A72

Experiment no - 02

Title - Implement Data types and Type casting.

1 Explain data types and storage classes.

- Each Variable in C has an associated data type. Each type requires amounts of memory and has some specific operation which can be performed over it. Let us briefly describe them one by one:
Following are the ~~examp~~ examples of some very common data types used in C:
 - Char - The most basic data types in C. It stores a single character and requires a single byte of memory in almost all compilers.
 - int - As the name suggests, an int variable is used to store an integer.
 - float - It is used to store decimal number with single precision.
 - double - It is used to store decimal with double precision.

Data types specific how we enter data into our programs & what types of data we enter. C language has some predefined set of data types to handle various kinds of data that we can use in our program. These datatypes have different storage capacities.

Storage classes: There are total four types of standard storage classes.

e.g. - auto, extern, static, register

2 Explain type casting

- Converting one datatype into another is known as type casting or type conversion for example if you want to store a long value into simple integer then you can type cast 'long' to 'int'. You can convert the values from one type to another explicitly using the cast operator as

follows - (type-name) expression.

Consider the following example of one integer variable by another to be performed as a floating point operation

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

```
main() {
    int sum = 17, count = 5;
    double mean;
    mean = (double)sum / count;
    printf("Value of mean : %.2f/n", mean);
}
```

Output -

Value of mean : 3.400000

It should be noted here that the cast operator has precedence over division, so the value of sum is first converted to type double & finally it gets divided by count yielding a double value.

Type conversions can be implicit which is performed by the compiler automatically or it can be specified explicitly through the use of the cast operator. Whenever type conversions are necessary, it is considered good programming practice to use the cast operator.

3 Strings, scope of variable, Tokens, Type casting.

→ Strings in C

In C programming, a string is a sequence of characters terminated with a null character \0. char c[] = "String"; When the compiler encounters a sequence of characters enclosed in the double quotation marks, it appends a null character \0 at the end by default.

Scope of Variable -

A scope is a region of the program, the scope of variables refers to the area of the program where the variables can be accessed after its declaration.

In C every variable defined in scope. In C variable can be declared in 3 places. These are:

local variables - Inside a function or block.

global variables - Out of all function

formal parameters - In the function parameters.

Tokens:-

Tokens in C is the most important element to be used in creating a program in C. We can define the token as the smallest individual element in C. For ex. ple, we cannot create a program in C without using tokens in C. Therefore, we can say that tokens in C is the building block or the basic component for creating a program in C language.

Tokens can be divided into the following categories:

1) Keywords in C 4) operators

2) Identifiers 5) constant

3) Strings 6) special characters

4. Write a program to calculate simple and compound interest

Program:-

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

```
int main()
{
```

```

int p, t;
Float r, si, amount, ci;

printf ("Please enter principle: ");
scanf ("%d, &p");

printf (" Please enter time: ");
scanf ("%d, &t");

printf ("Please enter interest: ");
scanf ("%F, &r);

si = p * t * r / 100;           // Simple interest formula

printf ("\nSimple interest = %.3F", si);

amount = p * pow((1+r/100), t); // Compound interest formula

ci = amount - p;

printf ("\nCompound interest = %.3F", ci);
return 0;
}

```

Output:-

Please enter principal : 6000
 Please enter time : 2
 Please enter interest : 6

Simple interest = 720.000

Compound interest = 741.600

5. Write a program to swap values of two variables with and without using third variable

→ With using third variable -

program:

```
# include<stdio.h>
int main(){
    double First, second, temp;
    pointF("Enter First number: ");
    scanf("%lf", &First);
    printF("Enter Second number: ");
    scanf("%lf", &second);
```

temp = First;

First = second;

Second = temp;

```
point ("\n After swapping First number = %.2lf\n", First);
point ("\n After Swapping Second number = %.2lf\n", Second);
return 0;
}
```

Output:-

Enter First number: 1.20

Enter Second number: 2.45

After Swapping first number: 2.45

After Swapping Second number: 1.20

Without using third variable

Program

```
#include <stdio.h>
int main() {
    double a, b;
    printf ("Enter a: ");
    scanf ("%lf", &a);
    printf ("Enter b: ");
    scanf ("%lf", &b);
```

$a = a - b;$

$b = a + b;$

$a = b - a;$

```
    print ("After swapping, a = %.2lf\n", a);
    print ("After swapping, b = %.2lf", b);
    return 0;
}
```

Output

Enter a : 10.25

Enter b : -12.5

After swapping, a = -12.50

After swapping, b = 10.25

6 Write a program to display the size of every data type using "size of" operator.

→ Program

```
#include <stdio.h>
int main()
{
    printf ("Size of char: %.lu\n", sizeof (char));
    printf ("Size of int: %.lu\n", sizeof (int));
    printf ("Size of float: %.lu\n", sizeof (float));
    printf ("Size of double: %.lu\n", sizeof (double));
    return 0;
}
```

Output:

Size of char: 1
 Size of int : 4
 Size of float : 4
 Size of double : 8

7 Write a program to illustrate the use of unary prefix & postfix increment and decrement operators.

→ Program

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

```
int main()
{
```

int Var1 = 5, Var2 = 5;

point F ("Before incrementing of A : %d\n", A);
 point F ("Before decrementing of B : %d\n", B);

point F ("After incrementing of A : %d\n", +A);
 point F ("After incrementing of B : %d\n", -B);

return 0;
 }

Output:

Before incrementing of A: 5
 Before decrementing of B: 5
 After incrementing of A = 6
 Before incrementing of B = 4

8 Write a program to input two number and display the maximum no.

→ Program:-

```
#include<stdio.h>
main()
{
    int a,b;
    point ("Enter two number :\n");
    scanf ("%d %d", &a, &b);
    if (a > b)
        pointF ("%d is a maximum number\n", a);
    else
        pointF ("%d is a maximum number\n", b);
```

```
return 0;
}
```

Output:

Enter two numbers:

25

40

40 is a maximum number

- 9 Write a program to find the largest of three numbers using ternary operators.

→ Program:-

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

```
int main()
```

```
int n1, n2, n3, man;
```

```
printf("Enter three numbers : ");
```

```
scanf ("%d %d %d", &n1, &n2, &n3);
```

//largest among n1, n2, n3.

```
man = (n1 > n2) ? (n1 > n3 ? n1 : n3) : (n2 > n3 ? n2 : n3);
```

```
printf("Largest number among %d, %d and %d is %d.", n1, n2, n3, man);
```

```
return 0;
```

```
}
```

Output:

Enter three number : 26

37

17

Largest number among 26, 37 & 17 is 37.

10 Write a program to find the roots of quadratic equation.

```
→ #include <math.h>
#include <stdio.h>
int main() {
    double a, b, c, discriminant, root1, root2, realPart, imagPart;
```

```
    printf("enter coefficients a, b and c: ");
    scanf("%lf %lf %lf", &a, &b, &c);
```

```
    discriminant = b * b - 4 * a * c;
```

// Condition for real & different roots

```
if (discriminant > 0) {
```

```
    root1 = (-b + sqrt(discriminant)) / (2 * a);
```

```
    root2 = (-b - sqrt(discriminant)) / (2 * a);
```

```
    printf("root1 = %.2f and root2 = %.2f", root1, root2);
```

}

// Condition for real and equal roots

```
else if (discriminant == 0) {
```

```
    root = root1 = root2 = -b / (2 * a);
```

```
    printf("root1 = root2 = %.2f", root);
```

}

// If roots are not real.

```
else {
```

```
    realPart = -b / (2 * a);
```

```

imagPart = sqrt (-discriminant) / (2 * a);
printf ("root1 = %.2f + %.2fi and root2 = %.2f - %.2fi, realPart, imagPart,
realPart, imagPart);

}

return 0;
}

```

Output:

Enter coefficients a,b and c : 3

2.5

6

root1 = -0.42 + 1.35i and root2 = -0.42 - 1.35i

- 11) Write a program to input name, marks of 5 subjects of a student & display the name of the student, the total marks scored, percentage scored and the class of result

→ #include <stdio.h>
include <string.h>

```

int main()
{
    int sl, english, maths, phy, che, cp, total;
    float per;
    char nm[20], div[10];
    printf("Enter Roll Number of the student : ");
    scanf ("%d", &sl);
    printf("Enter Name of the student : ");
    scanf ("%s", nm);

```

```

printf("Enter marks of English: ");
scanf("%d", &english);
printf("Enter marks of Maths: ");
scanf("%d", &maths);
printf("Enter marks of Physics: ");
scanf("%d", &phy);
printf("Enter marks of Chemistry: ");
scanf("%d", &che);
printf("Enter marks of Computer: ");
scanf("%d", &cp);
    
```

total = english + maths + phy + che + cp;

per = total / 5.0;

If (per >= 60)

strcpy(div, "First class");

else if (per < 60 && per >= 48)

strcpy(div, "Second");

else if (per < 48 && per >= 36)

strcpy(div, "Pass");

else

strcpy(div, "Fail");

printf("\n Roll No : %d \n Name of Student : %s \n", rl, nm);

printf(" Marks in English : %d \n Marks in Maths : %d \n", english, math);

printf(" Marks in Physics : %d \n Marks in Chemistry : %d \n", phy, chem);

printf(" Marks in Computer : %d \n", cp);

return 0;

}

Output

Enter Roll Number of the Student : 72

Enter Name of Student : pratik

Enter marks of English : 92

Enter marks of Maths : 89

Enter marks of Physics : 86

Enter marks of Chemistry : 93

Enter marks of Computer : 95

Roll No : 72

Name of Student : pratik

Marks in English : 92

Marks in Maths : 89

Marks in Physics : 86

Marks in Chemistry : 93

Marks in Computer : 95

Total marks = 455

Percentage = 91.00

Result : First class

Name - Pratik Jade
 Rollno - A72

Experiment no - 03

~~Ques.~~ Program or theory or flowchart

Title - Implement the Branch control statements inc

1 Write a program to count number of digits in given integer.

```
#include <stdio.h>
int main(){
    long long n;
    int count = 0;
    point F("Enter an integer: ");
    scanf ("%lld", &n);

    while (n!=0){
        n /= 10;
        ++count;
    }
    point ("Number of digits : %d", count);
}
```

Output:

```
Enter an integer : 346789
Number of digits : 6
```

2 Write a program to reverse a given integer.

→ # 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: 2345

Reversed number: 5432

4 Write a program to print the sum of digit of a number using for loop.

→ # include <stdio.h>

int main () {

 int n, i, sum=0

 printf ("Enter a positive integer: ");

 scanf ("%d", &n);

 for (i=1; i <= n; ++i) {

 sum += i;

}

```
    printf("Sum = %d", sum);
    return 0;
}
```

Output:

Enter a positive integer : 75
 Sum = 2850

5 Write a program to check whether a number is Palindrome or not.

```
→ #include <stdio.h>
Int main () {
    int n, reverseN=0, remainder, originalN;
    printf ("Enter an integer: ");
    scanf ("%d", &n);
    OriginalN=n;

    // // reversed integer is stored in reversedN
    while (n != 0) {
        remainder = n % 10;
        reverseN = reverseN * 10 + remainder;
        n /= 10;
    }

    // palindrome if originalN & reversedN are equal
    if (originalN == reverseN)
        printf ("%d is a palindrome.", originalN);
    else
        printf ("%d is not a palindrome.", originalN);
    return 0;
}
```

Output

Enter an integer : 1345

1345 is not a palindrome

6 Write a program to generate Fibonacci Series

→ #include <stdio.h>

int main()

int i, n, t1 = 0, t2 = 1, nextTerm;

printf("Enter the number of term: ");

scanf("%d", &n);

printf("Fibonacci Series: ");

for (i=1; i<=n; ++i) {

print("%d", t1);

nextTerm = t1 + t2;

t1 = t2

t2 = nextTerm;

}

return 0;

}

Output

Enter the number of terms : 10

Fibonacci Series : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

7 If a four-digit number is input through the keyboard, write a program to obtain the sum of the first & last digit of the number

→ #include <stdio.h>

int main()

{

int n, sum = 0, First digit, last digit;

printf("Enter number to find sum of first & last digit = ");
scanf("%d", &n);

// Find last digit of a number

lastdigit = n % 10;

// Find the first digit by dividing n by 10 until n greater than 10
while (n >= 10)

{

n = n / 10;

}

First digit = n;

// Sum of First & last digit

Sum = First digit + Last digit;

printf("Sum of first & last digit = %.d", sum);

return 0;

}

Output:

Enter number to find sum of first & last digit = 3456

Sum of first & last digit = 9

8 Write a program to find GCD (greater common divisor or HCF) & LCM (least common multiple) of two numbers.

→ #include <stdio.h>

int main() {

int a, b, x, y, t, gcd, lcm,

printf("Enter two integers\n");

scanf ("%d%d", &x, &y);

a = x;

b = y;

while (b != 0) {

 t = b;

 b = a % b;

 a = t;

}

gcd = a;

lcm = (x * y) / gcd;

printf("GCD of %d and %d = %d\n", x, y, gcd);

printf("LCM of %d and %d = %d\n", x, y, lcm);

return 0;

}

Output

Enter two integers 25, 30

GCD of 25 and 30 = 5

LCM of 25 and 30 = 150