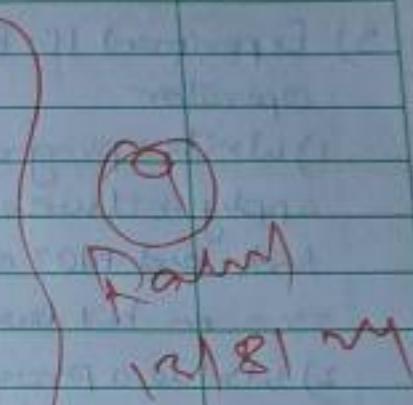


INDEX

Sr. No	Name of the Experiment	Page No	Date of Experiment	Date of Submission	Remarks
1.	Experiment 1: Installation, Environment, setup and starting in C language 1) Write a C program 01 to print "Hello world" 2) Write a C program 02 to print the address in multiple lines 3) Write a program 03 that prompts the user to enter their name and age 4) Write a C program 04 to add numbers, take number from user	1	6/08/25	12/8/25	
2.	Experiment 2: Operators 1) WAP a C program 05 to calculate the area and perimeter of a rectangle based on its length and width 2) WAP a program 06 to convert temp. from C to F	6/08/25	6/08/25		

INDEX

Sr. No	Name of the Experiment	Page No	Date of Experiment	Date of Submission	Remarks
3)	Experiment 11: Bitwise Operator	7-8	13/08/25	20/08/25	
	1) Write a program to apply bitwise OR, And and NOT operator -> on bit level.				
	2) Write a program to apply left shift and right shift opera - to r				
	Experiment 3 Condition and operators	9-10	13/08/25	20/08/25	
	1) WAP to take check by two angles is valid or not. If the validity is established, the checks, if the triangle is isosceles, Right angle or broken. Take sides of triangle as input from user				
	2) WAP to compute BMI index and print acc to given ranges.	10-13	13/09/25	20/09/25	
	3) WAP take input and check whether point are collinear	14-15	13/09/25	20/09/25	
	5) Take input and draw rectangle with oblique perimeter	16-17	13/09/25	20/09/25	
	4) WAP to find out what is the day on 1st January of this year		20/09/25	27/09/25	

Instructions

1. You should do the experiments.
2. You should understand the instruments.
3. Appendix
4. Procedure
5. All
6. Diagrams
7. Labels
8. Answers
- 9.

Experiment List

S.No Name:-

Page No.-

- | | | |
|----|--|------|
| 1) | Installation, Environment setup and starting with C language | 1-4 |
| 2) | Operators | 5-6 |
| 3) | Bitwise Operator (Experiment II) | 7-8 |
| 4) | Conditional Statement (Experiment 3-1) | 9-10 |
| 5) | Arrays | |
| 6) | Experiment - 6 Functions | |

Experiment 1:- Installation, Environment Setup and Starting with C language

1)

Q Write a C program to print "Hello World".

→ //Online C compiler to run C program online
#include <stdio.h>

```
int main() {  
    // Write C code here  
    printf("Hello World");  
  
    return 0;  
}
```

Input (Code)

Output

Hello World

... Code Execution Successful ...

→ Output Screen!

Output:-

Output:	(Ans)
UPES BIDHANI Dehradoon Uttarakhand School of Computer Science	...Code execution complete...

(Ans) Tugit

WPS O

brosiv allright

2)

O Write a C program to print the address in multiple lines (new lines).

Coding:-

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

```
int main () {
```

// Write a C code to print the address in multiple lines

```
printf (" UPES BIDHOLI\n");
```

```
printf (" Dehradoon Uttrakhand\n");
```

```
printf (" School of Computer science\n");
```

```
return 0;
```

```
}
```

→ Entered Code

Output

(Adults) student #

Please enter your name :- Pardeep

Please enter your age: 18

Hello, and it's

You are 18 years old

3.

E Write a program that prompts the user to enter their name and age.

→ Coding :-

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

```
int main( ) {
```

```
char *age;
```

```
// prompt for name
```

```
printf ("Please enter your name");
```

```
scanf ("%s", name);
```

```
// safer than gets()
```

```
// prompt for age
```

```
printf ("Please enter your age");
```

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

```
// output the result
```

```
printf ("Hello, %s You are %d years old\n", name, age);
```

```
return 0;
```

```
}
```

→ Entered code:-

Teacher's Signature : _____

- 4) Write a C program to add two numbers, take number from user.

Coding:-

Input code:-

#include <main (& studio.h)

int main() {

// Addition of two numbers

int a, b, sum;

printf ("The first number is \n");

scanf ("%d", &a);

printf ("The second number is \n");

scanf ("%d", &b);

sum = a + b;

printf ("Sum of the numbers is %d", sum);

return 0;

}

Teacher's Signature: _____

Output	(absolute) value is 77
The first number is	31 () now tell 41
The second number is	61 () which 61
The sum of the numbers is 100,	() together 100

Correct

Experiment 2: Operators

- 1) WAP a C program to calculate area and perimeter of a rectangle based on its length and width.

Coding :-

```
#include <stdio.h>
int main()
{
    float a, b, area, perimeter;
    printf("Length is \n");
    scanf("%f", &a);
    printf("breadth \n");
    scanf("%f", &b);
    area = a * b;
    perimeter = 2 * (a + b);
    printf("The area of the rectangle %f \n", area);
    printf("The Perimeter of rectangle %f \n", perimeter);
    return 0;
}
```

Entered code

Output

length is
40

breadth is
40

The area of rectangle is 1600.0000

The perimeter of rectangle 160.0000

2) Write a C program to convert temperature from Celsius to Fahrenheit using the formula $F = (C * 9/5) + 32$.

→ Coding :-

```
#include <stdio.h>
int main () {
    // Write code here
    float C,F;
    printf ("The temperature in Celsius\n");
    scanf ("%f", &C);
    F = (C * 9/5) + 32;
    printf ("The temperature in Fahrenheit %f\n", F);
    return 0;
}
```

↳ Entered code

Output

The temperature in Celsius is
40

The temperature in Fahrenheit is 104.00

Experiment - II

Bitwise operator

- 1) Write a program To apply bitwise OR, AND and NOT operators on bit level.

Input:-

→ #include <stdio.h>

```
int main() {  
    //Write C code here  
    int a = 5;  
    int b = 7;  
    printf("bitwise a or b is %d\n", a | b);  
    printf("bitwise a and b is %d\n", a & b);  
    printf("bitwise anot of a is %d\n", ~a);  
    return 0;  
}
```

→ Coding part

OUTPUT (variables) above #
for (i = 0; i < 4; i++)

bitwise a or b is 7

bitwise a and b is 2

bitwise not of a is -6

- -- Code execution begins here - 7

short break

2) Write a program to define after apply left shift and right shift operator.

→ Input:-

```
// Online C compiler to run (program online)
#include <stdio.h>
int main () {
    int a=5;
    int b=7;
    printf ("The left shift operator of %d\n", a<<1);
    printf ("The left shift operator of a %d\n", a<<2);
    printf ("The right shift operator of a %d\n", a>>1);
    printf ("The right shift operator of a %d\n", a>>2);
    return 0;
}
```

↳ Coding part

OUTPUT

The left shift operator of a is

The left shift operator of a is

The right shift operator of a is

The right shift operator of a is

TOP 3-1 Conditional Statement

- 1) WAP to take check if the triangle is valid or not
 If the validity is established then check which triangle and take input from user
 → Input

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

```
int main () {
```

```
float a, b, c;
```

```
// Input sides
```

```
printf ("Enter three sides of triangle . . . ");
```

```
scanf ("%f %f %f", &a, &b, &c);
```

```
// Check triangle validity
```

```
if ((a + b > c) && (a + c > b) && (b + c > a)) {
```

```
printf ("Triangle is valid \n");
```

```
// Check triangle type
```

```
if (a == b && b == c) {
```

```
printf ("It is an equilateral triangle \n");
```

```
{
```

```
else if (a == b || b == c || a == c) {
```

```
printf ("It is an isosceles triangle \n");
```

```
else {
```

```
print ("It is a scalene triangle \n"); }
```

{

```
// check triangle angle
```

```
if ((a*a == b*b + c*c) || (c*c == a*a + b*b))
```

{

```
print ("It is also a Right Angle. \n"); }
```

{ else {

```
print ("Triangle is not valid \n"); }
```

```
return 0; }
```

{

Algorithm:-

- Input three sides of triangle from the user
- Check validity using triangle Inequalities theorem
 - i. $a + b > c$, $a + c > b$, $b + c > a$
- If valid determine the type
 - All equal \Rightarrow Equilateral
 - Two equal \Rightarrow Isosceles
 - All different \Rightarrow Scalene
- Additionally, check if it satisfies the Pythagoras theorem for right angle triangle

True/False

OUTPUT

```
Enter three sides of triangle : 3 4 5  
Triangle is valid  
It is a scalene triangle  
It is also a right angled triangle
```

- 2) WAP to compute the BMI index of the person and print the BMI values as per the following ranges. You can use the following formula. $BMI = \text{weight} / (\text{height})^2$

	BMI
Starvation	<15
Anorexic	15.1 - 17.5
Under weight	17.6 to 18.5
Ideal	18.6 - 24.9
Over weight	25 to 29.9
Obese	30 - 39.9
Morbidity obese	40 and above

Input code:-

```
#include <stdio.h>
int main()
{
    float weight, height, bmi;
    // Inputs(1st one)
    printf("Enter your weight : ");
    scanf("%f", &weight);
```

// Input height

```
Printf ("Enter your height :");  
Scanf ("%f", &height);
```

// BMI formula

```
bmi = weight / (height * height)
```

// If else statement

```
If (bmi < 15) {
```

```
    Printf ("Category : Starvation\n"); }
```

```
else if (bmi >= 15.1 & bmi <= 17.5)
```

```
{    Printf ("Category : Anorexic\n"); }
```

```
else if (bmi >= 17.6 & bmi <= 18.5)
```

```
{    Printf ("Category : Underweight\n"); }
```

```
else if (bmi >= 18.6 & bmi <= 24.9)
```

```
{    Printf ("Category : Ideal\n"); }
```

```
else if (bmi >= 25 & bmi <= 25.9)
```

```
{    Printf ("Category : Overweight\n"); }
```

```
else if ( bmi >= 30 && bmi <= 39.9 )
    { printf ("Category: Obese \n"); }
else if ( bmi >= 40 )
    { printf ("Category : Morbidly obese \n"); }
else
    { printf ("Invalid BMI . range. \n"); }

return 0;
```

Algorithm:-

- Input the height and weight of a person
- Calculate the BMI of the person by the formula $BMI = \frac{W}{(H)^2}$
- Using if else statement determine the category the person falls in through the given table.

	BMI
Starvation	<15
Anorexic	15.1 - 17.5
Underweight	17.6 - 19.5
Ideal	19.6 - 24.9
Overweight	25 - 29.9
Obese	30 - 39.9
Morbidly obese	40 +

Output showing various methods

Output

Enter your weight : 90

Enter your height : 1.79

Category : Obese

90 kg

(obesity) about fit
In () more fit

- 3) WAP to check if three points (x_1, y_1) ; (x_2, y_2) and (x_3, y_3) are collinear or not.

Input code:-

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

```
int x1, y1, x2, y2, x3, y3, area;
```

// Input points

```
printf ("Coordinates of 1st point (x1,y1):");
scanf ("%d %d", &x1, &y1);
```

```
printf ("Coordinates of 2nd point (x2,y2):");
scanf ("%d %d", &x2, &y2);
```

```
printf ("Coordinates of 3rd point (x3,y3):");
scanf ("%d %d", &x3, &y3);
```

// Area of triangle formula

$$\text{area} = x^1 * (y^2 - y^3) + x^2 * (y^3 - y^1) + x^3 * (y^1 - y^2)$$

if (area == 0)

```
printf ("The points are collinear. \n");
```

Date _____

Expt. No. _____

Page No. 15

else

printf ("The points are Not collinear.\n");

return 0;

{

Output

Coordinate of 1st point (x_1, y_1) 1 1
Coordinate of 2nd point (x_2, y_2) 2 2
Coordinate of 3rd point (x_3, y_3) 3 3
The points are collinear.

Algorithm :-

- Input three points (x_1, y_1) , (x_2, y_2) , (x_3, y_3)
- Compute the value of area through given formula
$$\text{area} = x_1 \times (y_2 - y_3) + x_2 \times (y_3 - y_1) + x_3 \times (y_1 - y_2)$$
- Print "Collinear"
Else
- Print "Non Collinear"
- Stop.

Expt. No. _____

- Q5) WAP using ternary operator, user would input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangles should be three.

Input codes:-

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

```
int main () {
```

```
int l1, b1, l2, b2, l3, b3, P1, P2, P3, max;
```

// Input rectangles

```
printf ("Enter length and breadth of rectangle 1: ");
```

```
scanf ("%d %d", &l1, &b1);
```

```
printf ("Enter length and breadth of rectangle 2: ");
```

```
scanf ("%d %d", &l2, &b2);
```

```
printf ("Enter length and breadth of rectangle 3: ");
```

```
scanf ("%d %d", &l3, &b3);
```

// calculate perimeter

```
P1 = 2 * (l1 + b1);
```

Teacher's Signature _____

$$P_1 = 2 * (l_1 + b_1);$$

$$P_2 = 2 * (l_2 + b_2);$$

// max out of 3 perimeter

$$\max = (P_1 > P_2) ? ((P_1 > P_3) ? P_1 : P_3) : ((P_2 > P_3) ?$$

$$P_2 : P_3);$$

// printing results

```
printf("Perimeter of Rectangle 1 : %.d\n", P1);
printf("Perimeter of Rectangle 2 : %.d\n", P2);
printf("Perimeter of Rectangle 3 : %.d\n", P3);
printf("The rectangle with the highest perimeter has
      value perimeter = %.d\n", max);
```

return 0;

2

Output

Enter length and Breadth of Rectangle

-angle 1 : 4. 6

Enter length and Breadth of Rectangle

-angle 2 : 5 8

Enter length and Breadth of Rec

-angle 3 : 10 3

Perimeter of Rectangle 1 : 20

Perimeter of Rectangle 2 : 26

Perimeter of Rectangle 3 : 26

The rectangle with highest perim

-eter has ~~per~~ value = 26

Algorithm :-

- Take input of dimension of different Rectangles
- Compute the perimeter of each three Rectangles
 - $P_1 = 2 \times (l_1 + b_1)$
 - $P_2 = 2 \times (l_2 + b_2)$
 - $P_3 = 2 \times (l_3 + b_3)$
- Using if else statement determine no rectangle with highest perimeter
- Display the rectangle with highest perimeter
- Stop

Expt. No. _____

- 4) According to the gregorian calendar, it was Monday on the date 01/01/01. If any year is input to range by keyboard write a program to find out what is the day on 1st year of the year.

Input code:-

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

```
int main () {
```

```
    int year, total_days, day, i;
```

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

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

```
    total - days = 0;
```

```
    printf ("Enter Year : ");
```

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

```
    for (i = 1; i < year; i++) {
```

```
        if ((i % 4 == 0) && (i % 100 != 0) || (i % 400 == 0)) {
```

```
            total - days = total - days + 366;
```

 }

 else {

```
        total - days = total - days + 365;
```

 }

```
day = total - days / 7;  
if (day == 0) {  
    printf ("Monday"); } ?  
else if (day == 1) {  
    printf ("Tuesday"); } ?  
else if (day == 2) {  
    printf ("Wednesday"); } ?  
else if (day == 3) {  
    printf ("Thursday"); } ?  
else if (day == 4) {  
    printf ("Friday"); } ?  
else if (day == 5) {  
    printf ("Saturday"); } ?  
else {  
    printf ("Sunday"); }  
}  
return 0;
```

Output

Enter the year: 2025

Wednesday

global temp

(2025 global temp)

(avg 8.6 °C) last

(2025 global temp)

(avg 8.6 °C) last

(2025 global temp)

(avg 8.6 °C) last

36 + 5 * cap(1 - 1) / 100
= 36 + 5 * 0.88 = 43.4 °C

global temp - last = 7.4 °C

Algorithm:-

- Start
- Input the year
- Initialize for all years from 1 to year-1
 - If no year i is a leap year (i.e. $(i \% 400 == 0) \& (i \% 100 \neq 0) \& (i \% 4 == 0)$)
 - then add .366 to total - days
 - else, add .365 to total - days.
 - compute days = total - days * 1.7
 - check value of day and check if else statement
 - Output No day of 1st January of the entered year
 - Stop

Experiment-3-2 Loops

(Input)

1) ~~#include <stdio.h>~~

int main () {

int num;

int positive = 0, negative = 0, zero = 0;

int choice = 1;

while (choice == 1) {

printf ("Enter a number: ");

scanf ("%d", &num);

if (num > 0):

positive ++;

else if (num < 0)

negative ++;

else

zero ++;

printf ("Do you want to another number? (1 for yes,

Teacher's Signature: _____

```
0 j0n10 ); ");  
scanf ("%d", &choice);  
{  
    printf ("\n Total Positive numbers : %d\n", positive);  
    printf (" Total Negative number : %d\n", negative);  
    printf (" Total Zeroes : %d\n", zero);  
    return 0;  
}
```

Output

Enter a number : 900
Do you want to enter another number?
(1 for Yes, 0 for No)
Enter a number : 900
Do you want to enter another number?
(1 for Yes, 0 for No)
Total Positive number : 2
Total Negative number : 0
Total Zeros : 0

2)

Input

```
#include <stdio.h>

int main () {
    int num, i;

    printf ("Enter a number : ");
    scanf ("%d", &num);

    printf ("\n Multiplication table of %d :\n", num);
    for (i=1; i<10; i++) {
        printf ("%d * %d = %d\n", num, i, num*i);
    }
    return 0;
}
```

Ex

21

Output

Enter a number = 9

Multiplication Table of 9:

$$9 * 1 = 9$$

$$9 * 2 = 18$$

$$9 * 3 = 27$$

$$9 * 4 = 36$$

$$9 * 5 = 45$$

$$9 * 6 = 54$$

$$9 * 7 = 63$$

$$9 * 8 = 72$$

$$9 * 9 = 81$$

$$9 * 10 = 90$$

3)

Input

a)

#include <stdio.h>

int main()

int i, j, num = 1;

for (i = 1; i <= 3; i++) {

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

printf("%d ", num);

num++;

{

printf("\n");

return 0;

{

<u>Outow</u>
1
2 3
4 5 6

b)

Input

FFindude <stdio.h>

int main () {

int n = 5;

int i, j, num;

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

num = 1;

for (j = 0; j <= i; j++) {

printf("%d ", num);

num = num * (i - j) / (i + 1);

{

printf("\n");

{

return 0;

{

Output

1	1	1	1	1
1	2	1	1	1
1	3	3	1	1
1	4	6	4	1

4)

Input

#

Input code:-

#include <stdio.h>

int main() {

float population = 100000;

int year;

printf (" Population growth over 10 years : \n");

for (year = 1; year <= 10; year++) {

population = population + (population * 0.10);

printf (" Year %d : %.0f \n", year, population);

{

return 0;

}

Output

Population growth over 10 years;

Year 1 : 110000

Year 2 : 121000

Year 3 : 133100

Year 4 : 146410

Year 5 : 161051

Year 6 : 177156

Year 7 : 194872

Year 8 : 214359

Year 9 : 235795

Year 10 : 259374

5

Input code:-

#include <stdio.h>

int main()

int a, b, c, d;

int limit = 10000;

printf ("Ramanujan Numbers up to %d are :\n", limit)

for (a = 1; a*a*a < limit; a++)

for (b = a; b*b*b < limit; b++)

for (c = a; c*c*c < limit; c++)

for (d = c; d*d*d < limit; d++)

int sum1 = a*a*a + b*b*b;

int sum2 = c*c*c + d*d*d;

if (sum1 == sum2 & sum1 <= limit)

Teacher's Signature: _____

```
printf ("1. d = 1. d ^ 3 + 1. d ^ 3 + 1. d ^ 3 + 1. d ^ 3");
```

```
Sum 1, a, b, c, d );
```

{

}

{

{

{

return 0;

{

Output

Ramanujan Number up to 10000 are:

$$1729 = 1^3 + 12^3 = 9^3 + 10^3$$

$$4104 = 2^3 + 16^3 = 9^3 + 15^3$$

Experiment 4:- Variable and Scope of Variable

1) Input Code :-

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

```
int globalVar = 10;
```

```
void display();  
void modify();
```

```
int main () {
```

```
    printf ("Inside main(): globalVar = %d\n",  
           globalVar);
```

```
    display();  
    modify();
```

```
    printf ("After modify() call, globalVar = %d\n",  
           globalVar);
```

```
    return 0;
```

```
}
```

Void display () {

```
printf ("Inside display (): globalVar = %d\n", globalVar);
```

{

Void modify () {

```
globalVar = globalVar + 5;
```

```
printf ("Inside modify (): globalVar = %d\n", globalVar);
```

{

Output

Inside main() : globalVar = 10

Inside display() : globalVar = 10

Inside modify() : globalVar = 15

After modify() call , globalVar = 15

Expt. No. _____

2) Input code:-

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

```
int globalvar = 20;
```

```
void display () {
```

```
    int localvar = 10;
```

```
    printf ("Inside display ():\n");
```

```
    printf ("Local variable = %d\n", localvar);
```

```
    printf ("Global variable = %d\n", globalvar);
```

```
}
```

```
int main () {
```

```
    printf ("Inside main ():\n");
```

```
    printf ("Global variable = %d\n", globalvar);
```

```
    display();
```

Teacher's Signature : _____

Expt. No. _____

Date _____

Page No. 31

return 0;

S

Output :-

Inside main()

global variable = 20

Inside display():

local variable = 10

global variable = 20

3) Input code:-

#include <stdio.h>

int main()

{ int x = 10;

printf("Outside inner block : x = %d\n", x);

{

int y = 20;

printf("Inside inner block : x = %d, y = %d, z = %d\n", x, y);

{

int z = 30;

printf("Inside nested block : x = %d, y = %d, z = %d\n", x, y, z);

}

{

printf("Back in main block : x = %d\n", x);

return 0;

}

Output

Outside inner block : x 210

Inside inner block : x 210, y 220

Inside nested block : x 210, y 220, z 230

Back in main block : x 210

4

= Input code :-

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

```
void counter () {
```

```
    static int count=0;
```

```
    int normal = 0;
```

```
    count++;
```

```
    normal++;
```

```
    printf ("Static variable :: %d, Normal variable : %d\n", count, normal);
```

```
}
```

```
int main () {
```

```
    printf ("Calling counter() multiple times :\n");
```

```
    counter();
```

```
    counter();
```

```
    counter();
```

```
    return 0;
```

```
}
```

Output

Calling Counter() multiple times :

Static variable: 1, Normal Variable: 1

Static variable: 2, Normal Variable: 1

Static variable: 3, Normal Variable: 1

Experiment 5: Array :-

1) Input Code :-

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

```
int main() {  
    int n, i;  
    int arr[100];  
    int largest, secondlargest;
```

```
    printf("Enter the number of element : ");  
    scanf("%d", &n);
```

```
    printf("Enter %d integers :\n", n);  
    for (i = 0; i < n; i++) {  
        scanf("%d", &arr[i]);
```

{

```
    if (arr[0] > arr[1]) {  
        largest = arr[0];
```

```
        secondlargest = arr[1];
```

```
    } else {
```

largest = arr[4];

second largest = arr[0];

?

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

```
    if (arr[i] > largest) {
```

```
        second largest = largest;
```

```
        largest = arr[i];
```

```
    } else if (arr[i] > second largest && arr[i]
```

```
        != largest) {
```

?

```
printf("The second largest number is: %d\n",
```

```
second largest);
```

```
return 0;
```

?

Output

Enter the number of element : 3

Enter 3 integers :

12 30 80

The second largest number is : 30

2) Input Commands:-

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

```
int main() {
```

```
    int a[100], n, i, pos = 0, neg = 0, even = 0, odd = 0;
```

```
    printf("Enter number of elements: ");
```

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

```
    printf("Enter %d integers: \n", n);
```

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

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

```
        if (a[i] > 0) pos++;
```

```
        else if (a[i] < 0) neg++;
```

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

```
        else odd++;
```

```
}
```

```
    printf("Positive = %d, Negative = %d, Even = %d  
          Odd = %d", pos, neg, even, odd);
```

```
    return 0;
```

```
}
```

Output :-

Enter number of elements = 3

Enter 3 integers :

1 2 3

Positive . Negative = 0 Even = 1 Odd = 2

3 Input Commands:-

```
#include <stdio.h>
int main () {
    int arr [100], n, i, num, count = 0;
    printf ("Enter how many integers you want to enter:");
    scanf ("%d", &n);
    printf ("Enter %d integers:(n)", n);
    for (i=0; i<n; i++) {
        scanf ("%d", &arr [i]);
    }
    printf ("Enter the number to find its frequency:");
    scanf ("%d", &num);
    for (i=0; i<n; i++) {
        if (arr [i] == num) {
            count++;
        }
    }
    printf ("%d Frequency is %d in : %d\n", num, count);
    return 0;
}
```

Output :-

Enter how many integers you want to store : 6

Enter 6 integers :-

5

3

7

5

2

5

Enter the number to find its frequency = 5

frequency of 5 is : 3

4) Input for a C WAP program!

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

```
int main () {
```

```
    int A [20] [20], B [20] [20], C [20] [20];
```

```
    int m, n, P, Q;
```

```
    int i, j, k;
```

```
    printf ("Enter number of rows and columns of matrix A: ");
    scanf ("%d %d", &m, &n);
```

```
    printf ("Enter number of rows and columns of matrix B: ");
    scanf ("%d %d", &P, &Q);
```

```
    if (n != P) {
```

```
        printf ("Enter elements of Matrix B. not possible!\n");
```

```
        printf ("Columns of A (%d) must match rows of B(%d)\n"
                "n %d, P %d);\n", n, P);
```

```
    return 0;
```

```
}
```

```
    printf ("Enter elements of Matrix B (row-wise)\n"
            "n ");
```

```
    for (i = 0; i < P; i++) {
```

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

```
}
```

```
}
```

Teacher's Signature: _____

~~Step 1~~

```
printf ("Enter elements of Matrix B (row wise):n");
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
        scanf ("%d", &B[i][j]);
    }
}
```

3

```
for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++) {
        scanf ("%d", &c[i][j]);
        c[i][j] = 0;
    }
}
```

2
2

```
for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++) {
        for (k = 0; k < n; k++) {
            c[i][j] += A[i][k] * B[k][j];
        }
    }
}
```

3
2

```
printf ("\nMatrix C: \n");
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
        printf ("%d ", c[i][j]);
    }
}
```

2

printf ("\n");

Teacher's Signature:

```
printf ("\\n Matrix - B :\\n");  
for (i=0; i<P; i++) {  
    for (j=0; j<Q; j++) {  
        printf ("%4d", B[i][j]);  
    }  
    printf ("\\n");  
}
```

```
printf ("\\n Resultant Matrix (AxB) :\\n");  
for (i=0; i<m; i++) {  
    for (j=0; j<n; j++) {  
        printf ("%4d", c[i][j]);  
    }  
    printf ("\\n");  
}  
return 0;
```

Expt.

4)

Output :-

Enter number of rows and columns of Matrix A : 2 3
Enter number of rows and columns of matrix B : 3 2

Enter elements of Matrix A (row - wise)

1 2 3
4 5 6

Enter elements of Matrix B (row - wise)

7 8
9 10
11 12

Matrix A :

1 2 3
4 5 6

Matrix B :

7 8
9 10
11 12

Resultant matrix (A × B)

58 64
139 154

b) Functions

1) Input Programs

`#include <stdio.h>`

```
long long factR (int n) {
    if (n == 0) return 1;
    return n * factR (n - 1);
```

2

```
long long factNR (int n) {
```

```
    long long f = 1;
    for (int i = 1; i <= n; i++)
        f *= i;
    return f;
```

3

```
long long ncr (int n, int r) {
```

```
    return factR (n) / (factNR (r) * factNR (n - r));
```

4

```
int main () {
```

```
    int n, r;
```

```
    printf ("Enter n and r: ");

```

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

```
    if (r > n) {
```

```
        printf ("Invalid! r cannot be greater than n.\n");
```

```
    } else {
```

Teacher's Signature : _____

```
printf ("\\n C(%d,%d) = %d \\n", n, r, nCr(n,r));  
printf ("\\n Table of nCr values:\\n");  
for (int i = 0; i <= n; i++) {  
    for (int j = 0; j <= i; j++) {  
        printf ("%d %d \\n", i, j, nCr(i,j));  
    }  
}  
return 0;  
}
```

2) Write program:-

```
# include <stdio.h>

int GCD(int a, int b)
{
    if(b == 0)
        return a;
    else
        return GCD(b, a % b);
}

int main()
{
    int x, y;
    printf("Enter two integers: ");
    scanf("%d%d", &x, &y);
    printf("GCD of %d and %d is %d\n", x, y, GCD(x, y));
    return 0;
}
```

Out put :-

Enter two integers: 12 18
GCD of 12 and 18 is : 6

3) Input

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

```
int FIBO (int n) {
```

```
    if (n == 1) return n;
```

```
    return FIBO(n-1) + FIBO(n-2);
```

```
}
```

```
int main () {
```

```
    int num;
```

```
    printf ("Enter how many terms: ");
```

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

```
    printf ("Fibonacci sequence: \n");
```

```
    for (int i = 0; i < num; i++) {
```

```
        printf ("%d ", FIBO(i));
```

```
}
```

```
return 0;
```

```
}
```

Output :-

Enter how many terms: 10

Fibonacci sequence :

0 1 1 2 3 5 8 13 21 34

Expt. No. _____

4 Input Programming

```

#include <stdio.h>
int ISPRIME (int num) {
    if (num < 2) return 0;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) return 0;
    }
    return 1;
}
int main () {
    int start, end;
    printf ("Enter the range (start end): ");
    scanf ("%d %d", &start, &end);
    printf ("Prime number between %d and %d is :\n",
            start, end);
    for (int i = start; i <= end; i++) {
        if (ISPRIME (i)) {
            printf ("%d", i);
        }
    }
    return 0;
}

```

Teacher's Signature: _____

Output:-

Enter the range (start end) : 10 30

Prime numbers between 10 and 30 :

11 13 17 19 23 29