

# 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	6/08/25		
	1) Write a C program to print "Hello world"	01			
	2) Write a C program to print the address in multiple lines	02			
	3) Write a program that prompts the user to enter their name and age	03			
	4) Write a C program to add numbers, take number x from user	04			
2	Experiment 2: Operators	6/08/25	6/08/25		
	1) WAP a C program to calculate the area and perimeter of a rectangle based on its length and width	05			
	2) WAP a program to convert temp from C to F	06			

9  
Rahul  
12/8/24



# INDEX

Sr. No	Name of the Experiment	Page No	Date of Experiment	Date of Submission	Remarks
3)	Experiment 11: Bitwise Operator 1) Write a program to apply bitwise OR, And and NOT operators - to a bit level. 2) Write a program to apply left shift and right shift operators - to r	7-8	13/08/25	20/08/25	
	Experiment 3: Isosceles triangle operator -	9-10	13/08/25	20/08/25	
	1) WAP to take checky the triangle is valid or not. If the validity is established, the checks, if the triangle is isosceles, right angled or scalene. Take sides of triangle as a input from user				
2)	WAP to compute BMI index and print acc to given ranges.	10-12	13/08/25	20/08/25	
3)	WAP take input and check whether paint is	14-15	13/08/25	20/08/25	
5)	Take input and date & -mine the Rectangle with ob biggest perimeter	16-17	13/08/25	20/08/25	
4)	WAP to find out what is the day on 1st January of this year		20/08/25	27/08/25	

## Instructions

1. You sh experi
2. You s instru
3. Appo
4. Proc
5. All
6. Dis po
7. L
8. A
- 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 <sup>3-1</sup> )           | 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:

UPES BIDHOLI  
Dehradun Uttarakhand  
School of Computer Science

... Code Execution Complete ...

(Ans) Output

Output

Hello World



2)

Q Write a C Program to print the address in multiple line (new lines).

Soln:-

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

```
int main () {
```

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

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

```
printf(" Dehradun Uttarakhand\n");
```

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

```
return 0;
```

```
}
```

↳ Entered Code



Output

Please enter your name :- Pardeep  
Please enter your age: 18  
Hello, Pardeep  
You are 18 years old



3

Q 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);
```

```
    // after that
```

```
    // 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:-

→ Inbuilt code:-

```
#include <stdio.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 number is %d", sum);
```

```
return 0;
```

```
}
```

Teacher's Signature: \_\_\_\_\_



Output
The first number is 41
The second number is 61
The sum of the numbers is 100



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

Teacher's Signature : \_\_\_\_\_



Output
length is 40
breadth is 40
The area of rectangle is 1600.0000
The perimeter of rectangle 160.0000



2) WAD 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 is: \n");
    scanf ("%f", &C);
    F = (C * 9/5) + 32;
    printf ("The temperature in Fahrenheit is: %f \n", F);
    return 0;
}
```

↳ Entered code



## Output

The temperature in celsius is

40

The temperature in Fahrenheit is 104.00



Experiment - IIBitwise 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 a not of a is %d\n", ~a);

    return 0;

}

→ Coding part



OUTPUT (if available) absolute #

bitwise a or b is ?

bitwise a and b is ?

bitwise not of a.b - 6

--- Code execution completed ---



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

→ Input:-

```
//Online C compiler to run C 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

Code Execution Successful

Input printed



Exe 3-1 Conditional Statement

- 1) WAP to check if the triangle is valid or not.  
If the validity is established, triang 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 == c || b == c || a == b) {
```

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

```
else {
```



```
print f ("It is a scalene triangle \n");  
§  
// Check triangle angle  
if ((a*a == b*b + c*c) || (c*c == a*a + b*b))  
{  
    print f ("It is also a Right Angled. \n");  
} else {  
    print ("Triangle is not valid \n");  
}  
return 0;  
}
```



### Algorithm:-

- Input three sides of the triangle from the user
- Check validity using triangle inequality theorem

$$i.e. (a+b > c), (a+c > b), (b+c > a)$$

- 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 a right angle triangle



### OUTPUT

Enter three sides of triangle : 3 4 5

Triangle is valid

It is a scalene triangle

It is also a right angle 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:  $BMI = \frac{weight}{Height^2}$ ?

	BMI
Starvation	< 15
Anorexia	15.1 to 17.5
Underweight	17.6 to 18.5
Ideal	18.6 to 24.9
Overweight	25 to 29.9
Obese	30 - 39.9
Morbidly 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: Anorexia\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
Anorexia	15.1 - 17.5
Underweight	17.6 - 19.5
ideal	19.6 - 24.9
overweight	25 - 29.9
Obese	30 - 39.9
Morbidity obese	40 +



Output screen :

Output

Enter your weight : 90

Enter your height : 1.79

Category : Obese

Input Code

(doubt) double #

2 (1) more



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 (ask:-

```
#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
```

```
area = x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2);
```

```
if (area == 0)
```

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



else

printf("The points are Not collinear - N");

return 0;

§



### Output

Coordinate of 1<sup>st</sup> point ( $x_1, y_1$ ) 1 1

Coordinate of 2<sup>nd</sup> point ( $x_2, y_2$ ) 2 2

Coordinate of 3<sup>rd</sup> 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.



- Q5) WAP using ternary operator, the user should 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 Code:-

```
#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);
```



```
P2 = 2 * (l2 + b2);
```

```
P3 = 2 * (l3 + b3);
```

```
// max out of 3 perimeter
```

```
max = (P1 > P2) ? ((P1 > P3) ? P1 : P3) : ((P2 > P3) ?  
P2 : P3);
```

```
// 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;
```

```
}
```



### Output

Enter length and Breadth of Rectan

gle 1 : 4 6

Enter length and Breadth of Rectan

-gle 2 : 5 8

Enter length and Breadth of Rec

-tangle 3 : 10 3

Perimeter of Rectangle 1 : 20

Perimeter of Rectangle 2 : 26

Perimeter of Rectangle 3 : 26

The rectangle with high peri

-meter 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 the rectangle with highest perimeter
- Display the rectangle with highest perimeter
- Stop



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

```
{
```







### Algorithm:-

- Start
- Input the year
- Initialise for all years from 1 to year-1
  - If no year  $i$  is a leap year (i.e.  $(i \% 4 \neq 0 \text{ \& } i \% 100 \neq 0) \text{ \& } (i \% 400 \neq 0)$ )  
then add 365 to total - days
  - Else, add 366 to total - days
- Compute days = total - days + 1
- Check value of day and check if else statement
- Output the 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", &amp;num);

if (num > 0);  
positive ++;else if (num < 0)  
negative ++;

else

zero ++;

printf ("Do you want to enter number? (1 for yes, 0 for no)");

Teacher's Signature: \_\_\_\_\_



```
0 for No ) : " ) ;  
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



Input

2)

#include &lt;stdio.h&gt;

int main () {  
    int num, i;

printf ("Enter a number :");

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

printf ("\n Multiplication table of %d : \n", num);

for (i = 1; i &lt;= 10; i++) {

printf ("%d \* %d = %d \n", num, i, num\*i);

}

return 0;

}



### 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 &lt;stdio.h&gt;

int main() {

int i, j, num = 1;

for (i = 1; i &lt;= 3; i++) {

for (j = 1; j &lt;= 1; j++) {

printf("%d ", num);

num++;

}

printf("\n");

return 0;

}



<u>Out put</u>
1
2 3
4 5 6



b)

Input

#include &lt;stdio.h&gt;

int main() {

int n = 5;

int i, j, num;

for (i = 0; i &lt; n; i++) {

num = 1;

for (j = 0; j &lt; i; j++) {

printf("%d ", num);

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

}

printf("\n");

}

return 0;

}



Output

1

1

1

2

1

1

3

3

1

1

4

6

4

1



4)

Input

#

Input code:-

#include &lt;stdio.h&gt;

int main() {

float population = 100000;

int year;

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

for (year = 1; year &lt;= 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 &lt;stdio.h&gt;

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 &lt; limit; a++) {

for (b = a; b\*b\*b &lt; limit; b++) {

for (c = a; c\*c\*c &lt; limit; c++) {

for (d = c; d\*d\*d &lt; limit; d++) {

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

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

if (sum1 == sum2 &amp;&amp; sum1 &lt; limit) {

Teacher's Signature : \_\_\_\_\_



```
printf ("%d = %d ^ 3 + %d ^ 3 + %d ^ 3 + %d ^ 3",  
Sum, 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 &lt;stdio.h&gt;

int globalVar = 10;

void display();

void modify();

int main() {

printf("Inside 'main()': global var = %d\n",  
globalVar);

display();

modify();

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

return 0;

}



```
void display () {
```

```
    printf ("Inside display (): global var z = %d \n", globalvar);
```

```
}
```

```
void modify () {
```

```
    globalvar = globalvar + 5;
```

```
    printf ("Inside modify (): global var = %d \n", globalvar);
```

```
}
```



## Output

Inside main( ) : global Var = 10  
Inside display( ) : global Var = 10  
Inside modify( ) : global Var = 15  
After modify( ) call, global Var = 15



2) Input code:-

#include &lt;stdio.h&gt;

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();



Expt. No. \_\_\_\_\_

Date \_\_\_\_\_

Page No. 31

return 0;

\$



Output :-

Inside main()

Global variable = 20

Inside display():

Local variable = 10

Global variable = 20



3) Input code:-

#include &lt;stdio.h&gt;

int main() {

int x = 10;

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

{

int y = 20;

printf("Inside inner block: x = %d, y = %d, y = %d",  
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 :  $x = 10$

Inside inner block :  $x = 10, y = 20$

Inside nested block :  $x = 10, y = 20, z = 30$

Back in main block :  $x = 10$



4

=

Input code:-

#include &lt;stdio.h&gt;

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;

}

Teacher's Signature : \_\_\_\_\_



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[1];  
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) {
```

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

```
    }
```

```
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;
```

```
}
```



Out put :-

Enter number of elements : 3

Enter 3 integers :

1    2    3

Positive . Negative = 0    Even = 1    Odd = 2



3 Input Command:-

#include (stdio.h)

int main () {

int arr [100], n, i, num, count = 0;

printf ("Enter how many integers you want to store:");  
scanf ("%d", &n);

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

for (i = 0; i &lt; n; i++) {

scanf ("%d", &amp;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 ("Frequency is %d at : %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 the 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 ("n Enter elements of Matrix: not possible!\n");
```

```
printf ("Column of A (%d) must be equal rows of B (%d)  
      \n", n, p);
```

```
return 0;
```

```
}
```

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

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

```
scanf ("%d", &B[i][0]);
```

```
}
```

```
}
```

Teacher's Signature : \_\_\_\_\_



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

```
printf("\n 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]);
```

```
    }
}
```

```
for (i=0; i<m; i++) {
    for (j=0; j<n; j++) {
for (k=0; k<n; k++) C[i][j] = 0;
```

```
    }
}
```

```
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];
```

```
        }
    }
}
```

```
{
```

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

```
    }
}
```

```
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<Q; j++) {  
        printf("%4d", C[i][j]);  
    }  
    printf("\n");  
    return 0;  
}
```



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 x 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);
}

long long factNR(int n) {
    long long f = 1;
    for (int i = 1; i <= n; i++)
        f *= i;
    return f;
}

long long nCr(int n, int r) {
    return factR(n) / (factNR(r) * factNR(n-r));
}

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");
        return 0;
    }
```

Teacher's Signature : \_\_\_\_\_



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



2) Trut program:-

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

```
int gcd(int a, int b) {
```

```
if (b == 0)
```

```
return a;
```

```
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", x, y, gcd(x, y));
```

```
return 0;
```

```
}
```



Output :-

Enter two integers: 12 18

GCD of 12 and 18 is 6



3) Input

#include &lt;stdio.h&gt;

int FIBO (int n) {

if (n &lt;= 1) return n;

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

}

int main () {

int num;

printf ("Enter how many terms: ");

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

printf ("Fibonacci sequence: \n");

for (int i = 0; i &lt; 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. Inbuilt Program:-

```
#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 and 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