Output:

1
2 3
4 5 6

Expt. No. 3.2 4) Page No. 24 Date

4) The population of town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 years. Write a program to determine the population at the end of each year in last decade.

```
#include<stdio.h>
int main(){
    float population = 100000;
    int year;
    printf("year\t\tpopulation\n");
    for(year=1; year<=10; year++){
    population = population + (population * 0.10);
    printf("%d\t\t\t%.0f\n", year, population);
    }
    return 0;
}
```

Teacher's Signature

Output -

Year	Population
1	110000
2	121000
3	133100
4	146410
5	161051
6	177156
7	194872
8	214359
9	235795
100	2593794

Expt. No. 5 1) Page No. 25 Date

EXPERIMENT - 5

1) WAP to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers.

```
#include <stdio.h>
int main
int n, i, j, temp;
int arr[100];
printf("Enter number of elements: ");
Scanf("%d", &n);
printf("Enter %d integers :\n", n);
for (i=0; i<n ; i++) {
    Scanf("%d", &arr[i]);
}
for (i=0; i<n -1; i++){
    for(j=0; j<n -i-1 ; j++){
    if (arr[j] > arr[j+1]){
      temp = arr[j];
    arr[j] = arr[j+1];
    arr[j+1] = temp;
}
}
}
printf("\n the second largest number is: %d\n", arr[n-2]);
return 0;
```

Teacher's Signature

~~print~~

Output -

Enter the number of element : 5
Enter 5 integers
1 19 15 56 36
The second largest number is 35

Expt. No. 5 2) Page No. 26 Date

1) WAP to read a list of integer and store it in a single dimensional array. Write a C program to count and display positive, negative, odd and even numbers in an array.

```
#include<stdio.h>
int main(){
int pos=0, neg=0, odd=0, even=0;
int a[50], n, i;
printf("Enter number of element");
scanf("%d", &n);
printf("Enter %d integers\n", n);
for(i=0; i<n; i++){
  scanf("%d", &a[i]);
}
for(i=0; i<n; i++){
  if(a[i]>=0)
pos++
else
  neg++;
  if(a[i]%2==0)
    even++;
else
  oddd++;
}
```

Teacher's Signature

```
printf("positive numbers: %d\n", pos);
printf("negative number: %d\n", neg);
printf("even numbers: %d\n", even);
printf("odd numbers: %d\n", odd);
return 0;
}
```

Output:- enter number of element:-5
Enter integers :1, 3, 2, -4 5

pos = 3
neg = 2
even = 2
odd = 3

Expt. No. 5 3) Page No. 27 Date

3) WAP to read a list of integers and store it in a single dimensional array. Write a program to find the frequency of a particular number in a list of integers.

```
#include <stdio.h>
int main() {
int a[100], n, i, num, count=0;
printf("Enter the number of elements : ");
Scanf("%d", &n);
printf("Enter %d integers : \n", n);
for(i=0 ; i<n; i++) {
 Scanf("%d ", &a[i])}
}
printf("Enter the number whose frequency you
want to find:");
Scanf("%d", &num);
for(i=0; i<n; i++){
   if(a[i] == num)
      count++;
   }
printf("the frequency of %d is : %d \n", num,
                                        count);
return 0;
}.
```

Teacher's Signature

Output :- Enter the number of elements : 6

Enter 6 integer

1 1 3 3 4 1

Enter number whose frequency you want to find : 1

The frequency of 1 is 3

Expt. No. 54) Page No. 28 Date

4 WAP that reads two matrices A and B and computes the product A and B. Read matrix A and matrix B in row major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check ---- incompatability.

Teacher's Signature

Output :- Enter order of matrix A(m, n) = 2, 3
enter elements

1 2 3
4 5 6

Enter order of B matrix = 3, 2
Enter element

7 8
9 10
11 12

Resultant :-

58 64
139 154

Expt. No. 6) 1) Page No. 29 Date

EXPERIMENT - 6

1) Develop a recursive and non recursive fn FACT (num) to find the factorial of a no, n! defined by fact(n)=1 if n=0. otherwise FACT(n) = n* FACT(n-1). Using this fn, write a C program to compute - - - - - - . Suitable messages.

```
#include <stdio.h>
int main() {
int fact_recursive (int num)
{
if (num == 0)
    return 1;
else
return num * fact_recursive (num-1);
}

int fact_nonrecursive (int num)
{
  int f=1, i;
  for (i=1; i<=num; i++)
    f = f*i;
    return f;
}
```

Teacher's Signature

```
void main()
{
int n, r;
int nCr_rec, nCr_nonrec
printf("Enter values of n and r:");
Scanf("%d %d",&n, &r);
if (r>n)
{ printf("Invalid input");
n Cr_rec=(fact_recursive(n)/fact_non recursive(r)
                    * fact_nonrec(n-r);
printf("using recursive fn:");
printf("%d %d = %d\n", n, r, nCr_rec);
printf("using non recursive fn:");
printf("%d %d= %d\n", n, r, nCr_nonrec);
}
```

Output: Enter values of n and r : 6, 5
using recursive function;
(6,5) = 6

using non recursive fn-
(6,5) = 6

Expt. No. 6 2) Page No. 20 Date

2) Develop a recursive fn GCD(num1, num2) that accepts 2 integer arguments write a C program that invokes this fn to find greatest common divisor of 2 given integers

```
#include <stdio.h>
int gcd (int, int);
int main
{
  int a = 12
  int b = 18;
  int c = gcd(b, a);
  printf ("%d", c);
}

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

Teacher's Signature

Output :- 6

Expt. No. 6 3) Page No. 21 Date

3) Develop a recursive fn FIBO (num) that accepts an integer argument. Write a C program that invokes this fn to generate the Fibonacci sequence upto num

```
#include <stdio.h>
    int fibo (int num)
    {
    if (num == 0)
        return 0;
    else if (num == 1);
      return 1;
    else
      return fibo(num-1 + fibo(num-2);
    }
    void main ()
      {
      int n, i;
    printf ("Enter number of terms");
      scanf ("%d", &n);
      printf ("Fibonacci sequence upto %d terms
    for (i= 0; i<n ; i++)
      {
      printf ("%d", fibo (i));
      }
    }
```

Teacher's Signature _____

Output :- enter no of terms = 8

Fibonacci sequence upto 8 terms :-

0 1 1 2 3 5 8 13

Expt. No. 6 4) Page No. 32 Date

4) Develop a function ISPrime (num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this fn to generate prime number b/w the given ranges.

```
#include <stdio.h>
int ISPRIME (int num)
{
int i;
if (num <= 1)
return 0;
for (i=2; i<= num/2 ; i++)
  {
   if (num % i == 0)
     return 0;
  }
 return 1;
}
void main ()
{
int low, high, i;
printf ("\n Enter lower limits :");
```

Teacher's Signature

```
scanf("%d", low);
printf("Enter higher limits: ");
scanf("%d", &high);
printf("prime number b/w %d ad %d are
        : \n", low, high);
{
    if (ISPRIME (i))
        printf("%d", i);
}
}
```

Output :- Enter the lower limit :10
Enter the upper limit :30

Prime numbers b/w 10 and 30 are:-
11, 13, 17, 19, 23, 29

Expt. No. 8 Page No. 33 Date

EXPERIMENT - 8

Declare different types of pointer (int, float, char) and initialize them with the address of variable. Print the value of both the pointer and variables they point to.

```
#include <stdio.h>
    void main()
{
  int a=10
  float b=5.25;
    char c='A';
int *ptr1=&a;
float *ptr2=&b;
  float *ptr3=&c;
printf("values of variables:\n");
  printf("a= %d\n",a);
  printf("b= %f\n",b);
    printf("c= c\n",c);

printf("Addresses stored in pointer:\n");
printf("ptr1 = %p\n",ptr1);
printf("ptr2 = %p\n",ptr2);
```

Teacher's Signature

```
printf("ptr3 = %p\n\n", ptr3);

printf("values accessed using pointer :\n");
printf("*ptr1 = %d\n", *ptr1);
printf("*ptr2 = %.2f\n", *ptr2);
printf("*ptr3 = %c\n", *ptr3);
}
```

Output :- values of variables

a = 10
b = 5.25
c = A

Addresses of / Stored in pointers

ptr 1 = 0x7ffff68421a4
ptr 2 = 0x7ffff68421a8
ptr 3 = 0x7ffff68421ab

values accessed using pointer:

* ptr 1 = 10
* ptr 2 = 5.25
* ptr = A

Expt. No. 8 2) Page No. 34 Date

2) Perform pointer arithmetic on pointer of different data type. Observe how the memory addresses change and the effects on data access.

```
#include<stdio.h>
void main()
{
 int a=10
 float b=5.5;
  char c='A';
  int *p1=&a;
  float *p2=&b;
  char *p3=&c;
printf("Initial pointer addresses:\n");
 printf("*p1=%p\n",p1);
 printf("*p2=%p\n",p2);
 printf("*p3=%p\n\n",p3);
```

increment operators →

```
p1++
p2++
p3++
```

Teacher's Signature

```
printf("after incrementing pointers : \n");
printf("p1 = %p \n", p1);
printf("p2 = %p \n", p2);
printf("p3 = %p \n\n", p3);
```

decrement operator →

```
p1--
p2--
p3--
```

```
printf("after decrementing pointers : \n");
printf("p1 = %p \n", p1);
printf("p2 = %p\n", p2);
printf("p3 = %p\n", p3);
}
```

Expt. No. 4) 7) Page No. 35 Date

EXPERIMENT-4

1) Declare a global variable outside all fn's and use it inside various fn's & understand its accessibility.

```
#include<stdio.h>
void greet();
int a=20;
int main()
{
  printf("%d\n",a);
  greet();
  printf("%d\n",a);
  return 0;
}
void greet()
{
  void greet()
  {
    a=50;
  }
  printf("%d\n",a);
}
```

Teacher's Signature

OUTPUT →

20
50
50