

**FACULTY OF ENGINEERING &
TECHNOLOGY
PARUL INSTITUTE OF ENGINEERING
& TECHNOLOGY (DIPLOMA STUDIES)**



**COMPUTER ENGINEERING
DEPARTMENT
ADVANCED COMPUTER
PROGRAMMING
LAB
(03606152)**

SRNO	PRACTICAL LIST
1	Write a program to declare and print one dimensional array
2	Write a program to perform Insertion operation on 1-D array
3	Write a program to perform Deletion operation on 1-D array
4	Write a program to perform Searching operation on 1-D array
5	Write a program to perform Merging operation on 1-D array
6	Write a program to perform Sorting operation on 1-D array
7	Write a program to perform addition of 3x3 metrics using 2-D array
8	Write a program to perform two dimensional array to take input from user and display it on output screen
9	Write a program to show use of multi-dimensional array
10	Write programs to demonstrate use of different String functions like strlen(), strcpy, strcmp (), strlwr(), strupr(), strchr(), strcat().
11	Write programs to demonstrate use of common math and other functions like sqrt(), pow(), ceil(), round(), sin(), cos(), tan(), div(), abs() etc .
12	Write a program to perform addition using function with return type with arguments.
13	Write a program to perform addition using function with no return type and no arguments.
14	Write a program to show the use of call by value method.
15	Write a program to show the use of call by reference.
16	Write a program to print Fibonacci series using recursive function
17	Write a program to demonstrate the use of storage class
18	Write a program to perform factorial using recursive function.
19	Write a program to calculate the average of five numbers using function.
20	Write a Program to check whether a number is even or odd using function with arguments.
21	Write programs to demonstrate the use of Pointers.
22	Write programs to demonstrate use of array of Pointers and Pointers to array.
23	Write programs to demonstrate use of pointers of pointers.
24	Write programs to demonstrate use of Void pointers.
25	Write programs to demonstrate use of Null of pointers.

PRACTICAL 1

AIM: Write a program to declare and print one dimensional array.

PROGRAM:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i,a[5];
    printf("Enter 5 elements: \n");
    for (i=0; i < 5; i++)
    {
        scanf("%d ", &a[i]);
    }
    printf("Array Elements are: \n");
    for (i=0; i < 5; i++)
    {
        printf("a[%d] = %d\n",i, a[i]);
    }
    getch();
}
```

OUTPUT:

Enter 5 elements:

10
20
30
40
50

Array elements are:

a[0]=10
a[1]=20
a[2]=30
a[3]=40
a[4]=50

PRACTICAL 2

AIM: Write a program to perform Insertion operation on 1-D array.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10],position,i,n,value;
    printf("Enter number of elements in the array:");
    scanf("%d",&n);
    printf("Enter array elements:")
    for(i=0; i<n; i++)
    {
        scanf("%d",&a[i]);
    }
    printf("Enter the position");
    scanf("%d",& position);
    printf("Enter the value you want to insert:");
    scanf("%d",&value);
    for(i=n-1; i>position; i--)
    {
        a[i+1]=a[i];
    }
    a[position]=value;
    n=n+1;
    printf("\nResultant array is:");
    for(i=0; i<n; i++)
    {
        printf("a[%d] = %d\n",i,a[i]);
    }
    getch();
}
```

OUTPUT:

Enter number of elements in an array: 5

Enter array elements:

10

20

30

40

50

Enter the position: 2

Enter the value want to insert: 100

Resultant array is:

a[0]=10

a[1]=20

a[2]=100

a[3]=30

a[4]=40

a[5]=50

PRACTICAL 3

AIM: Write a program to perform Deletion operation on 1-D array.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10], index, i, n;
    printf("enter number of elements in array:");
    scanf("%d",&n);
    printf("enter array elements");
    for(i=0; i<n; i++)
    {
        scanf("%d",&a[i]);
    }
    printf("enter the index value where you want to delete element:");
    scanf("%d",&index);
    if(index>n)
    {
        printf("deletion not possible");
    }
    else
    {
        for(i=index; i<n; i++)
        {
            a[i]=a[i+1];
        }
    }
    printf("resultant array:");
    for(i=0; i<n; i++)
    {
        printf("%d",a[i]);
    }
}
getch();
}
```

OUTPUT:

Enter the number of elements in array: 4

Enter array element:

10

20

30

40

Enter the index value where you want to delete element: 2

Resultant array:

10

20

40

PRACTICAL 4

AIM: Write a program to perform Searching operation on 1-D array.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10], s, i, n;
    printf("enter number of elements in array\n");
    scanf("%d",&n);
    printf("enter array elements");
    {
        for(i=0; i<n; i++)
        {
            scanf("%d",&a[i]);
        }
        printf("enter a number to search:");
        scanf("%d",&s);
        for(i=0; i<n; i++)
        {
            if (a[i]==s)
            {
                printf("%d is present at location a[%d]",s,i);
                break;
            }
        }
        if(i==n)
        {
            printf("data not found");
        }
        getch();
    }
}
```

OUTPUT:

Enter the number of elements in array: 5

Enter array elements:

10

20

30

40

50

Enter a number to search: 20

20 is present at location a[1]

PRACTICAL 5

AIM: Write a program to perform Merging operation on 1-D array.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[20], b[20], i, n1, ,n2;
printf("Enter the number of elements in first array:");
scanf("%d",&n1);
printf("Enter number of elements in second array:");
scanf("%d",&n2);
printf("Enter first array elements:");
for(i=0; i<n1; i++)
{
scanf("%d",&a[i]);
}
printf("Enter second array elements:");
for(i=0; i<n2; i++)
{
scanf("%d",&b[i]);
}
n2=n1+n2;
for(i=0; i<=n2; i++)
{
a[n1+i] = b[i];
}
printf("Merge elements are:");
for(i=0; i<n2; i++)
{
printf("%d\n",a[i]);
}
getch();
}
```

OUTPUT:

```
Enter number of elements in first array: 3
Enter number of elements in second array: 4
Enter first array elements”
10
20
30
Enter second array elements:
40
50
60
70
Merge elements are:
10
20
30
40
50
60
70
```

PRACTICAL 6

AIM: Write a program to perform Sorting operation on 1-D array.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[20];
    int i; n; j; temp;
    printf("enter the size of array:");
    scanf("%d",&n);
    printf("enter array elements:");
    for(i=0; i<n; i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0; i<n; i++)
    {
        for(j=i+1; j<n; j++)
        {
            if(a[j] < a[i])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    printf("\n array in ascending order:");
    for(i=0; i<n; i++)
    {
        printf("%d\n",a[i]);
    }
    getch();
}
```

OUTPUT:

Enter the size of array: 3

Enter array elements:

30

10

20

Array in ascending order:

10

20

30

PRACTICAL 7

AIM: Write a program to perform addition of 3x3 metrics using 2-D array

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3][3], b[3][3], add[3][3]
int i,j;
clrscr();
printf("Enter Matrix a:");
for(i=0; i<3; i++)
{
for(j=0; j<3; j++)
{
scanf("%d", &a[i][j]);
}
}
printf("Enter Matrix b:");
for(i=0; i<3; i++)
{
for(j=0; j<3; j++)
{
scanf("%d", &b[i][j]);
}
}
for(i=0; i<3; i++) {
for(j=0; j<3; j++) {
add[i][j]=a[i][j]+b[i][j];
}
}
printf("\nAddition of a and b is: ");
for(i=0; i<3; i++) {
for(j=0; j<3; j++) {
printf(" %d ",add[i][j]);
}
printf("\n");
}
getch();
```

}

OUTPUT:

Enter Matrix A:

1 2 3
4 5 6
7 8 9

Enter Matrix B:

9 8 7
6 5 4
3 2 1

Addition of matrix A and B is:

10 10 10
10 10 10
10 10 10

PRACTICAL 8

AIM: Write a program to perform two dimensional array to take input from user and display it on output screen.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main() {
    int i,j;
    int a[2][3];
    printf("enter elements:");
    for(i=0; i<2; i++)
    {
        for(j=0; j<3; j++) {
            scanf("%d",&a[i][j]);
        }
    }
    printf("displaying array elements:");
    for(i=0; i<2; i++)
    {
        for(j=0; j<3; j++)
        {
            printf("a[%d][%d]=%d\n",i,j,a[i][j]);
        }
    }
    getch();
}
```

OUTPUT:

Enter elements:

10
20
30
40
50
60

Displaying array elements:

a[0][0]=10
a[0][1]=20
a[0][2]=30
a[1][0]=40
a[1][1]=50
a[1][2]=60

PRACTICAL 9

AIM: Write a program to show use of multi-dimensional array

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, j, k;
    int a[3][3][3]={
        {1,2,3},
        {4,5,6},
        {7,8,9}
    },
    {
        {10,11,12},
        {13,14,15},
        {16,17,18}
    },
    {
        {19,20,21},
        {22,23,24},
        {25,26,27}.
    }
};

printf("muulti-dimensional elements are:");
for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
    {
        for(k=0; k<3; k++)
        {
            printf("%d ",a[i][j][k]);
        }
        printf("\n");
    }
    printf("\n");
}
getch();
}
```

OUTPUT:

Multi-Dimensional elements are:

1 2 3

4 5 6

7 8 9

10 11 12

13 14 15

16 17 18

19 20 21

22 23 24

25 6 27

PRACTICAL 10

AIM: Write programs to demonstrate use of different String functions like **strlen()**, **strcpy**, **strcmp ()**, **strlwr()**,**strupr()**, **strchr()**, **strcat()**.

1. **strlen()** function

```
#include<stdio.h>
#include <string.h>
int main()
{
char ch[20]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
printf("Length of string is: %d",strlen(ch));
return 0;
}
```

OUTPUT:

Length of String is: 10

2. **strcpy()** function

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

int main( )
{
char source[ ] = "fresh2refresh" ;
char target[20]= "" ;
printf ( "\nsource string = %s", source ) ;
printf ( "\ntarget string = %s", target ) ;
strcpy ( target, source ) ;
printf ( "\ntarget string after strcpy( ) = %s", target ) ;
return 0;
}
```

OUTPUT:

```
source string = fresh2refresh
target string =
target string after strcpy( ) = fresh2refresh
```

3. strcmp() function

```
#include<stdio.h>
#include<string.h>
int main()
{
    char strg1[50], strg2[50];
    printf("Enter first string: ");
    gets(strg1);
    printf("Enter second string: ");
    gets(strg2);
    if(strcmp(strg1, strg2)==0)
    {
        printf("\nYou entered the same string two times");
    }
    else
    {
        printf("\nEnterd strings are not same!");
    }
    return 0;
}
```

OUTPUT:

```
Enter first string: compare
Enter second string: compare
You entered the same string two times
```

4. strupr() & strlwr() function

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[ ] = "Modify This String To Upper";
    printf("%s\n",strupr(str));
    printf("%s\n",strlwr(str));
    return 0;
}
```

OUTPUT:

Modify This String To Upper

Modify This String To Lower

5. strchr() function

```
#include<stdio.h>
#include <string.h>
int main()
{
    char mystr[30] = "I'm an example of function strchr";
    printf ("%s", strchr(mystr, 'f'));
    return 0;
}
```

OUTPUT:

f function s

6. strcat() function

```
#include<stdio.h>
#include <string.h>
int main()
{
    char s1[10] = "hello";
    char s2[10] = "world";
    strcat(s1,s2);
    printf("concatenation:%s",s1);
}
```

OUTPUT:

Concatenation: helloworld

PRACTICAL 11

AIM: Write programs to demonstrate use of common math and other functions like **sqrt()**, **pow()**, **ceil()**, **round()**, **sin()**, **cos()**, **tan()**, **div()**, **abs()** etc.

PROGRAM:

```
#include <stdio.h>
#include <math.h>
int main(){
    printf("%f\n",sqrt(10.0));
    printf("%f\n",exp(4.0));
    printf("%f\n",log(4.0));
    printf("%f\n",log10(100.0));
    printf("%f\n",fabs(-5.2));
    printf("%f\n",ceil(4.5));
    printf("%f\n",floor(-4.5));
    printf("%f\n",pow(4.0,.5));
    printf("%f\n",fmod(4.5,2.0));
    printf("%f\n",sin(0.0));
    printf("%f\n",cos(0.0));
    printf("%f\n",tan(0.0));
    return 0;}
```

OUTPUT:

```
3.162278
54.598150
1.386294
2.000000
5.200000
5.000000
-5.000000
2.000000
0.500000
0.000000
1.000000
0.000000
```

PRACTICAL 12

AIM: Write a program to perform addition using function with return type with arguments.

PROGRAM:

```
#include<stdio.h>
int main()
{
    int num1, num2, res;
    printf("\nEnter the two numbers : ");
    scanf("%d %d", &num1, &num2);
    //Call Function Sum With Two Parameters
    res = sum(num1, num2);
    printf("\nAddition of two number is : ");
    return (0);
}
int sum(int num1, int num2)
{
    int num3;
    num3 = num1 + num2;
    return (num3);
}
```

OUTPUT:

```
Enter the two numbers : 12 15
Addition of two number is : 27
```

PRACTICAL 13

AIM: Write a program to perform addition using function with no return type and no arguments.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
//Function Declaration
void add();
void main()
{
    //Function Calling
    add();
    getch();
}

//Function Definition
void add()
{
    int a,b,c;
    printf("\nEnter The Value of A & B :");
    scanf("%d%d",&a,&b);
    c=a+b;
    printf("\nTotal : %d",c);
}
```

OUTPUT:

Enter The Value of A & B : 5 & 2
Total:7

PRACTICAL 14

AIM: Write a program to show the use of call by value method.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void swap(int a, int b)
{
    int temp;
    temp=a;
    a=b;
    b=temp;
}
void main()
{
    int a=100, b=200;
    clrscr();
    swap(a, b); // passing value to function
    printf("\nValue of a: %d",a);
    printf("\nValue of b: %d",b);
    getch();
}
```

OUTPUT:

Value of a: 200

Value of b: 100

PRACTICAL 15

AIM: Write a program to show the use of call by reference method.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void swap(int *a, int *b)
{
    int temp;
    temp=*a;
    *a=*b;
    *b=temp;
}
void main()
{
    int a=100, b=200;
    clrscr();
    swap(&a, &b); // passing value to function
    printf("\nValue of a: %d",a);
    printf("\nValue of b: %d",b);
    getch();
}
```

OUTPUT:

Value of a: 200

Value of b: 100

PRACTICAL 16

AIM: Write a program to print Fibonacci series using recursive function

PROGRAM:

```
#include<stdio.h>
void printFibonacci(int n){
    static int n1=0,n2=1,n3;
    if(n>0){
        n3 = n1 + n2;
        n1 = n2;
        n2 = n3;
        printf("%d ",n3);
        printFibonacci(n-1);
    }
}
int main(){
    int n;
    printf("Enter the number of elements: ");
    scanf("%d",&n);
    printf("Fibonacci Series: ");
    printf("%d %d ",0,1);
    printFibonacci(n-2); //n-2 because 2 numbers are already printed
    return 0;
}
```

OUTPUT:

Fibonacci Series: 15
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

PRACTICAL 17

AIM: Write a program to demonstrate the use of storage class.

- Auto
- External
- Static
- Register

PROGRAMS:

- **Auto:**

```
#include <stdio.h>
int main()
{
    int a; //auto
    char b;
    float c;
    printf("%d %c %f",a,b,c); // printing initial default value of automatic variables a, b, and c.
    return 0;
}
```

OUTPUT:

garbage garbage garbage

- **Static:**

```
#include<stdio.h>
static char c;
static int i;
static float f;
static char s[100];
void main ()
{
printf("%d %d %f %s",c,i,f); // the initial default value of c, i, and f will be
printed.
}
```

OUTPUT:

0 0 0.000000 (null)

- **Register:**

• 0 0 0.000000 (null)

```
#include <stdio.h>
int main()
{
register int a; // variable a is allocated memory in the CPU register. The initia
l default value of a is 0.
printf("%d",a);
}
```

OUTPUT: 0

- **Extern**

```
#include <stdio.h>
int a;
int main()
{
extern int a; // variable a is defined globally, the memory will not be allocat
ed to a
printf("%d",a);
}
```

OUTPUT: 0

PRACTICAL 18

AIM: Write a program to print factorial using recursive function

PROGRAM:

```
#include<stdio.h>
// recursive function to find factorial of a number
int factorial(int n)
{
    if(n!=0)
        return n*factorial(n-1); // general case
    else
        return 1; // base case
}
int main()
{
    int num, result;
    printf("Enter a positive number: ");
    scanf("%d",&num);
    result= factorial(num); //function call
    printf("Result = %d\n",result);
    return 0;
}
```

OUTPUT:

```
Enter a positive number: 5
Result= 120
```

PRACTICAL 19

AIM: Write a program to calculate the average of five numbers using function.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n1,n2,n3,n4,n5;
    void avg (int a, int b, int c, int d, int e);
    clrscr();
    printf("Enter five numbers:");
    scanf("%d %d %d%d %d ",&n1,&n2,&n3,&n4,&n5);
    avg(n1,n2,n3,n4,n5);
    getch();
}
void avg (int a, int b, int c, int d, int e)
{
    float average;
    average=(a+b+c+d+e)/5.0;
    printf("Average=%f",average);
}
```

OUTPUT:

```
Enter five numbers: 10 10 10 10 10
Average:10
```

PRACTICAL 20

AIM: Write a Program to check whether a number is even or odd using function with arguments.

PROGRAM:

```
#include <stdio.h>
#include <stdlib.h>
int find_Num(int); //function prototype
int main()
{
    int num;
    printf("Enter a number to check odd or even\n");
    scanf("%d", &num);
    find_Num(num); //calling the function
    getch();
    return 0;
}
//create function
int find_Num(int num){ //function definition
    if (num % 2 == 0){
        printf("\n%d is an even number", num);
    }
    else{
        printf("\n%d is an odd number", num);
    }
}
```

OUTPUT:

Enter a number to check odd or even

46

46 is an even numbers

PRACTICAL 21

AIM: Write programs to demonstrate the use of Pointers.

PROGRAM:

```
#include <stdio.h>
int main()
{
    int num = 10;
    int * ptr;
    /* Stores the address of num to pointer type */
    ptr = &num;
    printf("Address of num = %d\n", &num);
    printf("Value of num = %d\n", num);
    printf("Address of ptr = %d\n", &ptr);
    printf("Value of ptr = %d\n", ptr);
    printf("Value pointed by ptr = %d\n", *ptr);
    return 0;
}
```

OUTPUT:

```
Address of num = 6356748.
Value of num = 10
Address of ptr = 6356744
Value of ptr = 6356748
Value pointed by ptr = 10
```

PRACTICAL 22

AIM: Write programs to demonstrate use of array of Pointers and Pointers to array.

PROGRAM 1: (Array of pointers)

```
#include <stdio.h>

const int MAX = 3;

int main () {

    int var[] = { 10, 100, 200};
    int i, *ptr[MAX];

    for ( i = 0; i < MAX; i++) {
        ptr[i] = &var[i]; /* assign the address of integer. */
    }

    for ( i = 0; i < MAX; i++) {
        printf("Value of var[%d] = %d\n", i, *ptr[i] );
    }

    return 0;
}
```

OUTPUT:

```
Value of var[0] = 10
Value of var[1] = 100
Value of var[2] = 200
```

PROGRAM 2: (pointers to aarray)

```
#include<stdio.h>
void main()
{
    int a[3] = {1, 2, 3};
    int *p = a;
    for (int i = 0; i < 3; i++)
    {
        printf("%d", *p);
        p++;
    }
    return 0;
}
```

OUTPUT:

1 2 3

PRACTICAL 23

AIM: Write programs to demonstrate the use of Pointers to pointer.

PROGRAM:

```
#include <stdio.h>

// C program to demonstrate pointer to pointer
int main()
{
    int var = 789;

    // pointer for var
    int *ptr2;

    // double pointer for ptr2
    int **ptr1;

    // storing address of var in ptr2
    ptr2 = &var;

    // Storing address of ptr2 in ptr1
    ptr1 = &ptr2;

    // Displaying value of var using
    // both single and double pointers
    printf("Value of var = %d\n", var );
    printf("Value of var using single pointer = %d\n", *ptr2 );
    printf("Value of var using double pointer = %d\n", **ptr1);

    return 0;
}
```

OUTPUT:

Value of var = 789
Value of var using single pointer = 789
Value of var using double pointer = 789

PRACTICAL 24

AIM: Write programs to demonstrate the use of Void pointer.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a=10;
void *p1=&a;
float f=2.5;
void *p2=&f;
printf("Value of a is : %d\n",*(int *)p1);
printf("Address of a is:%x\n",p1);
printf("Value of f is : %f\n",*(float *)p2);
printf("Address of f is: %x",p2);
getch();
}
```

OUTPUT:

```
Value of a is : 10
Address of a is:fff4
Value of f is : 2.50000
Address of f is: fff0
```

PRACTICAL 25

AIM: Write programs to demonstrate the use of Null pointer.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int *p=NULL;
printf("The Value of p is : %x",p);
getch();
}
```

OUTPUT:

The Value of p is : 0