PROGRAMING FOR PROBLEM SOLVING LAB PRACTICAL

MY PROGRAMS

My Details:-

Name- Navjot Singh
CRN- 1921070
Branch- I.T.
Year- 1st
Submitted to- prof. Kamaljeet Kaur Dhillon

1. Hello Budding Engineers

```
    #include<stdio.h>
     int main()
     {
      puts("Hello Budding Engineers\n");
     return 0;
     }
```

OUTPUT:

> Hello Budding Engineers

2. Address using puts

```
#include<stdio.h>
 int main()
  {
     puts("My address:");
  puts(Sangrur,Punjab, India");
return 0;
}
```

OUTPUT:

> My address:
> Sangrur, Punjab, India

3. Sum of two numbers

```
 #include<stdio.h>
  int main()
  {
       int a, b, sum;
     printf("Enter two numbers\n");
  scanf("%d %d",&a,&b);
  sum=a+b;
    printf("sum=%d\n",sum);
     return 0;
```

```
    }
```

OUTPUT:

> Enter two numbers
> 12
> 26
> sum=38

4. Convert Celsius to Fahrniet

```c
#include<stdio.h>

int main()
 {
   float fahr, cel;
   printf("Enter the temperature in celsius: ");
   scanf("%f", &cel);

   fahr = (1.8 * cel) + 32.0;
   printf("\nTemperature in Fahrenheit: %.2f F\n", fahr);

   return 0;
   }
```

OUTPUT:

> Enter the temperature in celsius: 32
>
> Temperature in Fahrenheit: 89.60 F

5. Multiplication Table

```c
#include<stdio.h>
int main()
{
int num,n, i,table;
printf("Enter a number");
scanf("%d",&num);
printf("Enter the number upto which you wanna see the table\n");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
table=0;
table=num*i;
printf("%d*%d=%d\n",num,i,table);
}
return 0;
}
```

OUTPUT:

> Enter a number 7
> Enter the number upto which you wanna see the table
> 10
> *71=7*
> *72*=14

> 7 3=21
> 7 4=28
> 7 5=35
> 7 6=42
> 7 7=49
> 7 8=56
> 7 9=63
> 7 10=70

6. Perimeter and area of circle

```
#include<stdio.h>
#define PI 3.14
int main()
{
float radius,area,peri;
printf("Enter the radius of circle\n");
scanf("%f",&radius);
area=PI*radius*radius;
peri=2*PI*radius;
printf("Area of the circle=%f\n",area);
printf("Perimeter of the circle=%f\n",peri);
return 0;
}
```

OUTPUT:

> Enter the radius of circle
> 5
> Area of the circle=78.500000
> Perimeter of the circle=31.400000

7. Reverse

```
#include <stdio.h>
int main()
{
    int n, reversedNumber = 0, remainder;
    printf("Enter an integer: ");
    scanf("%d", &n);
    while(n != 0)
    {
        remainder = n%10;
        reversedNumber = reversedNumber*10 + remainder;
        n /= 10;
    }
    printf("Reversed Number = %d", reversedNumber);
    return 0;
}
```

OUTPUT:

> Enter an integer: 1356
> Reversed Number = 6531

8. Swapping without using a third variable

```c
    #include <stdio.h>

int main()
{
  int x, y, t;

  printf("Enter two integers\n");
  scanf("%d%d", &x, &y);

  printf("Before Swapping\nFirst integer = %d\nSecond integer = %d\n", x, y);

  t = x;
  x = y;
  y = t;

  printf("After Swapping\nFirst integer = %d\nSecond integer = %d\n", x, y);

  return 0;
}
```

OUTPUT:

> Enter an integer: 1356
> Reversed Number = 6531

9. Even Odd

```c
    #include <stdio.h>
int main()
{
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);

    if(number % 2 == 0)
        printf("%d is even.", number);
    else
        printf("%d is odd.", number);
    return 0;
}
```

OUTPUT:

> Enter an integer: 13
> 13 is odd.

10. Factorial

```c
    #include<stdio.h>
int main()
{
 int i,fact=1,number;
 printf("Enter a number: ");
  scanf("%d",&number);
    for(i=1;i<=number;i++){
      fact=fact*i;
  }
```

```
  printf("Factorial of %d is: %d",number,fact);
return 0;
}
```

OUTPUT:

> Enter a number: 3
> Factorial of 3 is: 6

11. Weekdays using switch case

```
#include <stdio.h>

  int main()
    {

int week;

printf("Enter week number(1-7): ");
scanf("%d", &week);

switch(week)
{
    case 1:
        printf("Monday");
        break;
    case 2:
        printf("Tuesday");
        break;
    case 3:
        printf("Wednesday");
        break;
    case 4:
        printf("Thursday");
        break;
    case 5:
        printf("Friday");
        break;
    case 6:
        printf("Saturday");
        break;
    case 7:
        printf("Sunday");
        break;
    default:
        printf("Invalid input! Please enter week number between 1-7.");
}

  return 0;
}
```

OUTPUT:

> Enter week number(1-7): 4
> Thursday

12. Arithmetic operations using switch case

```c
#include<stdio.h>

void main()
{
    int a,b;
    int op;

    printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n");
    printf("Enter the values of a & b: ");
    scanf("%d %d",&a,&b);
    printf("Enter your Choice : ");
    scanf("%d",&op);
    switch(op)
    {
    case 1  :
        printf("Sum of %d and %d is : %d",a,b,a+b);
        break;
    case 2  :
        printf("Difference of %d and %d is : %d",a,b,a-b);
        break;
    case 3  :
        printf("Multiplication of %d and %d is : %d",a,b,a*b);
        break;
    case 4  :
        printf("Division of Two Numbers is %d : ",a/b);
        break;
    default :
        printf(" Enter Your Correct Choice.");
        break;
    }

}
```

OUTPUT:

> 1.Addition
> 2.Subtraction
> 3.Multiplication
> 4.Division
> Enter the values of a & b: 12
> 76
> Enter your Choice : 3
> Multiplication of 12 and 76 is : 912

13. Prime Numbers

```c
#include <stdio.h>
int main()
{
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    for(i = 2; i <= n/2; ++i)
    {

        if(n%i == 0)
        {
```

```
            flag = 1;
            break;
        }
    }
    if (n == 1)
    {
      printf("1 is neither a prime nor a composite number.");
    }
    else
    {
        if (flag == 0)
          printf("%d is a prime number.", n);
        else
          printf("%d is not a prime number.", n);
    }
    return 0;
    }
```

OUTPUT:

> Enter a positive integer: 13
> 13 is a prime number.

14. Fibonacci Series

```
    #include <stdio.h>
int main()
{
    int prev=0;
    int curr=1;
    int n;
    int next,a;
    printf("Enter the number of terms\n");
  scanf("%d", &n);

  printf("First %d terms of Fibonacci series are:\n",n);

  for (a = 0; a < n; a++)
  {
    if (a <= 1)
      next = a;
    else
    {
      next = prev + curr;
      prev = curr;
      curr = next;
    }
    printf("%d\n", next);
  }

  return 0;
}
```

OUTPUT:

> Enter the number of terms
> 8
> First 8 terms of Fibonacci series are:

```
0
1
1
2
3
5
8
13
```

15. Palindrome

```c
#include <stdio.h>
int main()
{
    int n, reversedInteger = 0, remainder, originalInteger;
    printf("Enter an integer: ");
    scanf("%d", &n);
    originalInteger = n;
    while( n!=0 )
    {
        remainder = n%10;
        reversedInteger = reversedInteger*10 + remainder;
        n /= 10;
    }
    if (originalInteger == reversedInteger)
        printf("%d is a palindrome.", originalInteger);
    else
        printf("%d is not a palindrome.", originalInteger);

    return 0;
}
```

OUTPUT:

```
Enter an integer: 12321
12321 is a palindrome.
```

16. Palindrome words

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

void check(char [], int);

int main()
{
    char word[15];

    printf("Enter a word to check if it is a palindrome\n");
    scanf("%s", word);
    check(word, 0);

    return 0;
}

void check(char word[], int index)
{
```

```
    int len = strlen(word) - (index + 1);
    if (word[index] == word[len])
    {
        if (index + 1 == len || index == len)
        {
            printf("The entered word is a palindrome\n");
            return;
        }
        check(word, index + 1);
    }
    else
    {
        printf("The entered word is not a palindrome\n");
    }
}
```

OUTPUT:

> Enter a word to check if it is a palindrome
> naman
> The entered word is a palindrome

17. Star Half Pyramid

```
    #include <stdio.h>
int main()
{
    int x, y, rows;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(x=1; x<=rows; ++x)
    {
        for(y=1; y<=x; ++y)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

> Enter number of rows: 5
>
> *

18. Star Full Pyramid

```
#include <stdio.h>
int main()
{
    int i, j, rows;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
```

```
    for(i=rows; i>=1; --i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

> Enter number of rows: 8
> *
> * *
> ** *
> ** **
> * ** **
>
> ---
>
> ---
>
> ---

19. Star Inverted Half Pyramid

```
#include <stdio.h>
int main()
{
    int i, j, rows;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=rows; i>=1; --i)
    {
        for(j=1; j<=i; ++j)
        {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

OUTPUT:

> Enter number of rows: 4
>
> ---
>
> ---
>
> - ◦
> -

20. 1D Array

```
    #include <stdio.h>

  void  main()
{
```

```
    int arr[10];
    int i;
       printf("\n\nRead and Print elements of an array:\n");
       printf("----------------------------------------\n");

    printf("Input 10 elements in the array :\n");
    for(i=0; i<10; i++)
    {
        printf("element - %d : ",i);
        scanf("%d", &arr[i]);
    }

    printf("\nElements in array are: ");
    for(i=0; i<10; i++)
    {
        printf("%d  ", arr[i]);
    }
    printf("\n");
}
```

OUTPUT:

> # Read and Print elements of an array:
>
> Input 10 elements in the array :
> element - 0 : 1
> element - 1 : 3
> element - 2 : 4
> element - 3 : 2
> element - 4 : 6
> element - 5 : 8
> element - 6 : 5
> element - 7 : 7
> element - 8 : 0
> element - 9 : 9
>
> Elements in array are: 1 3 4 2 6 8 5 7 0 9

21. Maximum Size of an array

```
#include <stdio.h>

#define MAX_SIZE 100  // Maximum array size

int main()
{
    int arr[MAX_SIZE];
    int size, i, toSearch, found;

    printf("Enter size of array: ");
    scanf("%d", &size);


    printf("Enter elements in array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
```

```
    }

    printf("\nEnter element to search: ");
    scanf("%d", &toSearch);


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

        if(arr[i] == toSearch)
        {
            found = 1;
            break;
        }
    }

    if(found == 1)
    {
        printf("\n%d is found at position %d", toSearch, i + 1);
    }
    else
    {
        printf("\n%d is not found in the array", toSearch);
    }

    return 0;
}
```

OUTPUT:

> Enter the number of elements in array
> 4
> Enter 4 integers
> 2
> 4
> 0
> 9
> Maximum element is present at location 4 and it’s value is 9.

22. 2D Array

```
#include<stdio.h>
int main(){
   /* 2D array declaration*/
   int disp[2][3];
   /*Counter variables for the loop*/
   int i, j;
   for(i=0; i<2; i++) {
      for(j=0;j<3;j++) {
         printf("Enter value for disp[%d][%d]:", i, j);
         scanf("%d", &disp[i][j]);
      }
   }

   printf("Two Dimensional array elements:\n");
   for(i=0; i<2; i++) {
      for(j=0;j<3;j++) {
         printf("%d ", disp[i][j]);
```

```
         if(j==2){
            printf("\n");
         }
      }
   }
   return 0;
}
```

OUTPUT:

> Enter value for disp[0][0]:2
> Enter value for disp[0][1]:3
> Enter value for disp[0][2]:2
> Enter value for disp[1][0]:3
> Enter value for disp[1][1]:4
> Enter value for disp[1][2]:2
> Two Dimensional array elements:
> 2 3 2
> 3 4 2

23. Sum of two matrices

```
#include <stdio.h>

int main()
{
   int m, n, c, d, first[10][10], second[10][10], sum[10][10];

   printf("Enter the number of rows and columns of matrix\n");
   scanf("%d%d", &m, &n);
   printf("Enter the elements of first matrix\n");

   for (c = 0; c < m; c++)
      for (d = 0; d < n; d++)
         scanf("%d", &first[c][d]);

   printf("Enter the elements of second matrix\n");

   for (c = 0; c < m; c++)
      for (d = 0 ; d < n; d++)
         scanf("%d", &second[c][d]);

   printf("Sum of entered matrices:-\n");

   for (c = 0; c < m; c++) {
      for (d = 0 ; d < n; d++) {
         sum[c][d] = first[c][d] + second[c][d];
         printf("%d\t", sum[c][d]);
      }
      printf("\n");
   }

   return 0;
}
```

OUTPUT:

> Enter the number of rows and columns of matrix
> 2
> 3
> Enter the elements of first matrix
> 1
> 2
> 3
> 6
> 4
> 8
> Enter the elements of second matrix
> 2
> 4
> 7
> 2
> 5
> 7
> Sum of entered matrices:-
> 3 6 10
> 8 9 15

24. Transpose of matrix

```
    #include<stdio.h>
void main()
{
int a[10][10], b[10][10];
int m,n,i,j;
printf("Enter size of matrix A as m, n:");
scanf("%d%d",&m,&n);
printf("\n Enter elements of matrix A row wise\n",m ,n);
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
scanf("%d",&a[i][j]);
}
}
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
b[j][i]=a[i][j];
}
}
printf("\n\nTranspose of matrix is:\n");
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
{
printf("%d",b[i][j]);
}
}
}
```

OUTPUT:

```
Enter size of matrix A as m, n:3
2

Enter elements of matrix A row wise
1
4
2
5
7
9

Transpose of matrix is:
127459
```

25. Substraction of two matrices

```c
#include <stdio.h>

int main()
{
   int m, n, c, d, first[10][10], second[10][10], difference[10][10];

   printf("Enter the number of rows and columns of matrix\n");
   scanf("%d%d", &m, &n);
   printf("Enter the elements of first matrix\n");

   for (c = 0; c < m; c++)
     for (d = 0 ; d < n; d++)
       scanf("%d", &first[c][d]);

   printf("Enter the elements of second matrix\n");

   for (c = 0; c < m; c++)
     for (d = 0; d < n; d++)
         scanf("%d", &second[c][d]);

   printf("Difference of entered matrices:-\n");

   for (c = 0; c < m; c++) {
     for (d = 0; d < n; d++) {
       difference[c][d] = first[c][d] - second[c][d];
       printf("%d\t",difference[c][d]);
     }
     printf("\n");
   }

   return 0;
}
```

OUTPUT:

```
Enter the number of rows and columns of matrix
2
2
Enter the elements of first matrix
3
2
```

> 4
> 5
> Enter the elements of second matrix
>
> 7
> 9
> 0
> 2
> Difference of entered matrices:-
> -4 -7
> 4 3

26. Multiplication of two matrices

```
    #include <stdio.h>

int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];

  printf("Enter number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter elements of first matrix\n");

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
      scanf("%d", &first[c][d]);

  printf("Enter number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);

  if (n != p)
    printf("The matrices can't be multiplied with each other.\n");
  else
  {
    printf("Enter elements of second matrix\n");

    for (c = 0; c < p; c++)
      for (d = 0; d < q; d++)
        scanf("%d", &second[c][d]);

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++) {
        for (k = 0; k < p; k++) {
          sum = sum + first[c][k]*second[k][d];
        }

        multiply[c][d] = sum;
        sum = 0;
      }
    }

    printf("Product of the matrices:\n");

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++)
        printf("%d\t", multiply[c][d]);
```

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

  return 0;
}
```

OUTPUT:

> Enter number of rows and columns of first matrix
> 2
> 2
> Enter elements of first matrix
> 23
> 65
> 98
> 10
> Enter number of rows and columns of second matrix
> 2
> 2
> Enter elements of second matrix
> 54
> 60
> 0
> 160
> Product of the matrices:
> 1242 11780
> 5292 7480

27. Square of a number using function

```
     #include<stdio.h>

int square(int); // function prototype declaration.

void main()
{
     int number, answer;

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

     answer = square(number);  //Call function.

     printf("Square of %d is %d.", number, answer);
}

int square(int n)
{

     return(n*n);
}
```

OUTPUT:

> Enter your number:6
> Square of 6 is 36.

28. Swaping call by value

```c
#include <stdio.h>


void swap(int, int);

int main()
{
   int x, y;

   printf("Enter the value of x and y\n");
   scanf("%d%d",&x,&y);

   printf("Before Swapping\nx = %d\ny = %d\n", x, y);

   swap(x, y);

   printf("After Swapping\nx = %d\ny = %d\n", x, y);

   return 0;
}

void swap(int a, int b)
{
   int temp;

   temp = b;
   b = a;
   a = temp;
    printf("Values of a and b is %d  %d\n",a,b);
}
```

OUTPUT:

> Enter the value of x and y
> 5
> 8
> Before Swapping
> x = 5
> y = 8
> Values of a and b is 8 5
> After Swapping
> x = 5
> y = 8

29. Swapping call by reference

```c
    #include <stdio.h>
    void swap(int *n1, int *n2);
    int main()
    {
        int num1, num2;
printf("Enter the num1 and num2");
```

```
scanf("%d%d",&num1,&num2);
        swap( &num1, &num2);
        printf("num1 = %d\n", num1);
        printf("num2 = %d", num2);
        return 0;
    }
    void swap(int* n1, int* n2)
    {
        int temp;
        temp = *n1;
        *n1 = *n2;
        *n2 = temp;
    }
```

OUTPUT:

> Enter the num1 and num2
> 3
> 5
> num1 = 5
> num2 = 3

30. Factorial using recursion

```
    #include<stdio.h>
int main()
{
    #include <stdio.h>
    long int multiplyNumbers(int n);
    int main()
    {
        int n;
        printf("Enter a positive integer: ");
        scanf("%d", &n);
        printf("Factorial of %d = %ld", n, multiplyNumbers(n));
        return 0;
    }
    long int multiplyNumbers(int n)
    {
        if (n >= 1)
            return n*multiplyNumbers(n-1);
        else
            return 1;
    }
```

OUTPUT:

> Enter a positive integer: 5
> Factorial of 5 = 120

31. Fibonacci series using recursion

```
    #include<stdio.h>

    int Fibonacci(int);

    int main()
```

```
    {
       int n, i = 0, c;

       scanf("%d",&n);

       printf("Fibonacci series\n");

       for ( c = 1 ; c <= n ; c++ )
       {
          printf("%d\n", Fibonacci(i));
          i++;
       }

       return 0;
    }

    int Fibonacci(int n)
    {
       if ( n == 0 )
          return 0;
       else if ( n == 1 )
          return 1;
       else
          return ( Fibonacci(n-1) + Fibonacci(n-2) );
    }
```

OUTPUT:

> 4
> Fibonacci series
> 0
> 1
> 1
> 2

32. Structure

```
    #include <stdio.h>
struct student
{
    char name[50];
    int roll;
    float marks;
} s[10];
int main()
{
    int i;
    printf("Enter information of students:\n");
    // storing information
    for(i=0; i<10; ++i)
    {
        s[i].roll = i+1;
        printf("\nFor 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");
```

```
    }
    printf("Displaying Information:\n\n");
    // displaying information
    for(i=0; i<10; ++i)
    {
        printf("\nRoll number: %d\n",i+1);
        printf("Name: ");
        puts(s[i].name);
        printf("Marks: %.1f",s[i].marks);
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

> Roll number: 5
> Name: bhola
> Marks: 32.0
>
> Roll number: 6
> Name: kaka
> Marks: 66.0
>
> Roll number: 7
> Name: billa
> Marks: 100.0
>
> Roll number: 8
> Name: john
> Marks: 89.0
>
> Roll number: 9
> Name: jacky
> Marks: 38.0
>
> Roll number: 10
> Name: Blacky
> Marks: 42.0

33. Pointers

```
#include<stdio.h>
int main()
{
int a,*p;
a=10;
p=&a;
printf("%d\n",p);
printf("%d\n",*p);
printf("%d\n",&p);
return 0;
}
```

OUTPUT:

> -1306961924

```
10
-1306961936
```

34. Addition using Pointers

```c
#include <stdio.h>

int main()
{
   int first, second, *p, *q, sum;

   printf("Enter two integers to add\n");
   scanf("%d%d", &first, &second);

   p = &first;
   q = &second;

   sum = *p + *q;

   printf("Sum of the numbers = %d\n", sum);

   return 0;
}
```

OUTPUT:

```
Enter two integers to add
4
65
Sum of the numbers = 69
```

35. Pointers to an array

```c
#include<stdio.h>

int main()
{
  int arr[5] = { 1, 2, 3, 4, 5 };
  int *ptr = arr;

  printf("%p\n", ptr);
  return 0;
}
```

OUTPUT:

```
0x7ffd4b542230
```

36. Pointers to a function

```c
#include <stdio.h>

void fun(int a)
{
    printf("Value of a is %d\n", a);
}
```

```
int main()
{
    void (*fun_ptr)(int) = fun;

    fun_ptr(10);

    return 0;
}
```

OUTPUT:

> Value of a is 10

37. Printing values of an array using pointers

```
      #include<stdio.h>
void main()
{
    int a[5]= {5,4,6,8,9};
    int *p=&a[0];
    int i;

    for(i=0; i<5; i++)
        printf("\nArray[%d] is %d ",i,*(p+i));
    for(i=0; i<5; i++)
        printf("\n %d at %u ",*(p+i),(p+i));
  }
```

OUTPUT:

> Enter two integers to add
> 58
> 94
> Sum of the numbers = 152

38. Bubble Sort

```
#include <stdio.h>

int main()
{
  int array[100], n, c, d, swap;

  printf("Enter number of elements\n");
  scanf("%d", &n);

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

  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);

  for (c = 0 ; c < n - 1; c++)
  {
    for (d = 0 ; d < n - c - 1; d++)
    {
      if (array[d] > array[d+1]) /* For decreasing order use < */
      {
        swap       = array[d];
```

```
        array[d]   = array[d+1];
        array[d+1] = swap;
      }
    }
  }

  printf("Sorted list in ascending order:\n");

  for (c = 0; c < n; c++)
     printf("%d\n", array[c]);

  return 0;
}
```

OUTPUT:

> Enter number of elements
> 5
> Enter 5 integers
> 23
> 4
> 54
> 87
> 98
> Sorted list in ascending order:
> 4
> 23
> 54
> 87
> 98

39. Quick Sort Using Recursion

```
    #include <stdio.h>



    void quicksort (int [], int, int);



      int main()

{

    int list[50];

    int size, i;



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

    scanf("%d", &size);

    printf("Enter the elements to be sorted:\n");
```

```
    for (i = 0; i < size; i++)
    {
        scanf("%d", &list[i]);
    }
    quicksort(list, 0, size - 1);
    printf("After applying quick sort\n");
    for (i = 0; i < size; i++)
    {
        printf("%d ", list[i]);
    }
    printf("\n");

    return 0;
}
void quicksort(int list[], int low, int high)
{
int pivot, i, j, temp;
if (low < high)
{
    pivot = low;
    i = low;
    j = high;
    while (i < j)
    {
        while (list[i] <= list[pivot] && i <= high)
        {
            i++;
        }
        while (list[j] > list[pivot] && j >= low)
        {
```

```
            j--;

        }

        if (i < j)

        {

            temp = list[i];

            list[i] = list[j];

            list[j] = temp;

        }

    }

    temp = list[j];

    list[j] = list[pivot];

    list[pivot] = temp;

    quicksort(list, low, j - 1);

    quicksort(list, j + 1, high);

}

}
```

OUTPUT:

> Enter the number of elements: 5
> Enter the elements to be sorted:
> 45
> 32
> 76
> 455
> 34
> After applying quick sort
> 32 34 45 76 455