ABES Engineering College, Ghaziabad

Affiliated to AKTU Lucknow
Department of CSE-DS

Lab Manual

Session 2022-23 (Odd Semester)

Name	:	PAVIT SAXENA
Subject Name	:	Data Structures using C
Subject Code	:	KCS-351
Branch	:	CSE(DS)
Section	:	CSDS-B

ABES Engineering College, Ghaziabad
B.Tech(CSE-DS) III Semester
Session 2022-23 (Odd Semester)

Course Title: DS LAB (KCS-351)   Name of Faculty: Mr. Amit Kumar Pandey

List of Experiments

1. (a) Write a program in C to find the average marks of a class of students.

1. (b) Write a program in C to find the largest number from given n numbers in array

2. Write a program in C to implement addition and multiplication of two 2D arrays.

3. Write a program in C to transpose a 2D array..

4. Write a program in C to implement stack using array.

5. Write a program in C to implement queue using array.

6. Write a program in C to implement circular queue using array.

7. Write a program to implement singly linked list.

8. Write a program in C to implement stack using linked list.

9. Write a program in C to implement queue using linked list.

10. Write a program in C to implement circular queue using linked list.

11. Write a program in C to implement binary tree using linked list.

12. Write a program in C to implement binary search tree using linked list.

13. Write a program in C to implement tree traversal using linked list.

14. Write a program in C to implement BFS using linked list

15. Write a program in C to implement DFS using linked list

16. Write a program in C to implement Linear Search.

17. Write a program in C to implement Binary Search.

18. Write a program in C to implement Bubble Sorting.

19. Write a program in C to implement Selection Sorting.

20. Write a program in C to implement Insertion Sorting.

21. Write a program in C to implement Merge Sorting.

22. Write a program in C to implement Heap Sorting.

23.Write a Program for polynomial addition using Linked List.

24.Write a Program for Infix to Postfix Conversion.

(Faculty signature) Mr. Prabhat Singh, Incharge(CSE-DS)

Program 1 (a):

Write a program in C to find the AVERAGE MARKS of a class of students. Suppose that there are n students in the class.

Objective of program1(a) : To make students familiar with the concept of arrays and provide them with hands on practice.

Algorithm 1:

Let n be the number of students in the class.

Step 1: Get the value of n as input from user.

Step 2: Initialise sum=0, i=1.

Step 3: Repeat Step 4, 5 and 6 for (i<=n)

Step 4: Get the marks of student i from user.

Step 5: sum=sum+marks[i].

Step 6: Set i=i+1.

Step 7: Calculate the average as sum/n.

Step 8: Print the value of average.

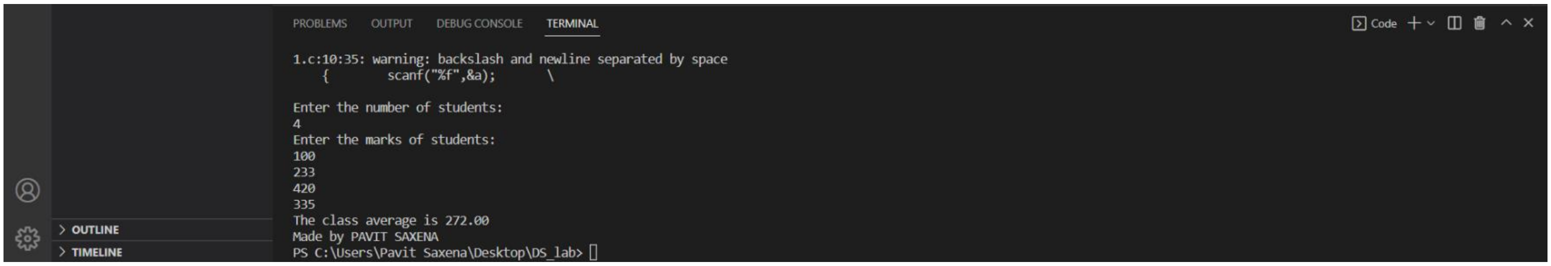
Step 9: Exit.

CODE-

```
#include <stdio.h> int

main()

{   int n;     float a,i,sum=0;      float average;

printf("Enter the number of students: \n");

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

```
 printf("Enter the marks of students:\n");
for(i=0;i<n;i++){        scanf("%f",&a);       \
sum=sum+a;
} average=sum/n;      printf("The class average is
%.2f   \n",average);   printf("Made   by   PAVIT
SAXENA ");
 return 0;
}
```

OUTPUT:



```
1.c:10:35: warning: backslash and newline separated by space
    {        scanf("%f",&a);       \

Enter the number of students:
4
Enter the marks of students:
100
233
420
335
The class average is 272.00
Made by PAVIT SAXENA
PS C:\Users\Pavit Saxena\Desktop\DS_lab> 
```

Program 1 (b):

Write a program in C to find the LARGEST NUMBER from given n numbers.

Algorithm 2:

Let n be total numbers.

Step 1: Get the value of n as input from user.

Step 2: For (i<=n), get the number from user and Set i=i+1.

Step 3: Initialise big = a[1].

Step 4: Initialise i =2 and repeat step 5and 6 for (i<=n).

Step 5: if (a[i] > big), Set big = a[i].

Step 6: Set i = i+1.

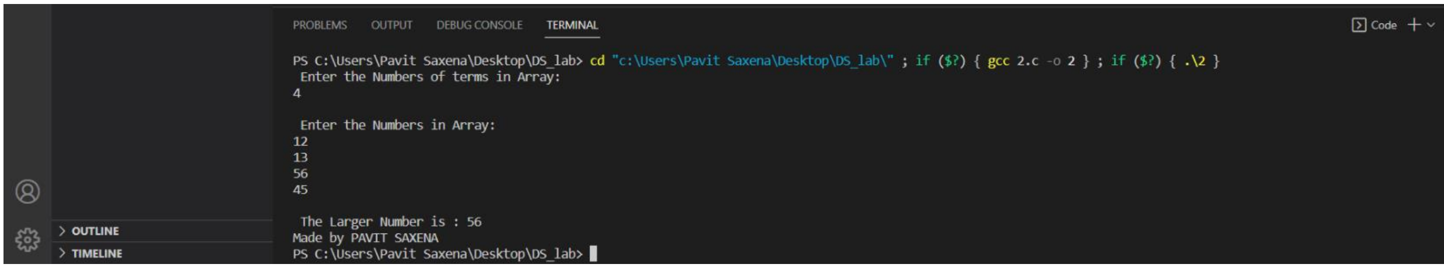
Step 7: Print the value of big.

Step 8: Exit.

CODE:

```
# include <stdio.h> int  main( )  {  int  a[20], n, i,
lar=0  ;   printf("  Enter  the  Numbers  of  terms  in
Array:\n")
; scanf("%d",&n);  printf("\n Enter the
Numbers in Array: \n") ;  for (  i=0 ; i<n-1 ;
i++)
{  scanf("%d
",&a[i]) ;  } for (
i=0 ; i<n-1 ; i++)
{ if ( a[i] > lar ) { lar = a[i] ;  }}  printf("\n
The Larger Number is : %d\n",lar) ;
printf("Made by PAVIT SAXENA ");


return  0 ;
}
```

OUTPUT:



```
PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Desktop\DS_lab\" ; if ($?) { gcc 2.c -o 2 } ; if ($?) { .\2 }
 Enter the Numbers of terms in Array:
4

 Enter the Numbers in Array:
12
13
56
45

 The Larger Number is : 56
Made by PAVIT SAXENA
PS C:\Users\Pavit Saxena\Desktop\DS_lab> █
```

Program 2:

Write a program in C to implement ADDITION AND MULTIPLICATION OF TWO ARRAYS:

Addition of Two Arrays

Algorithm 2:

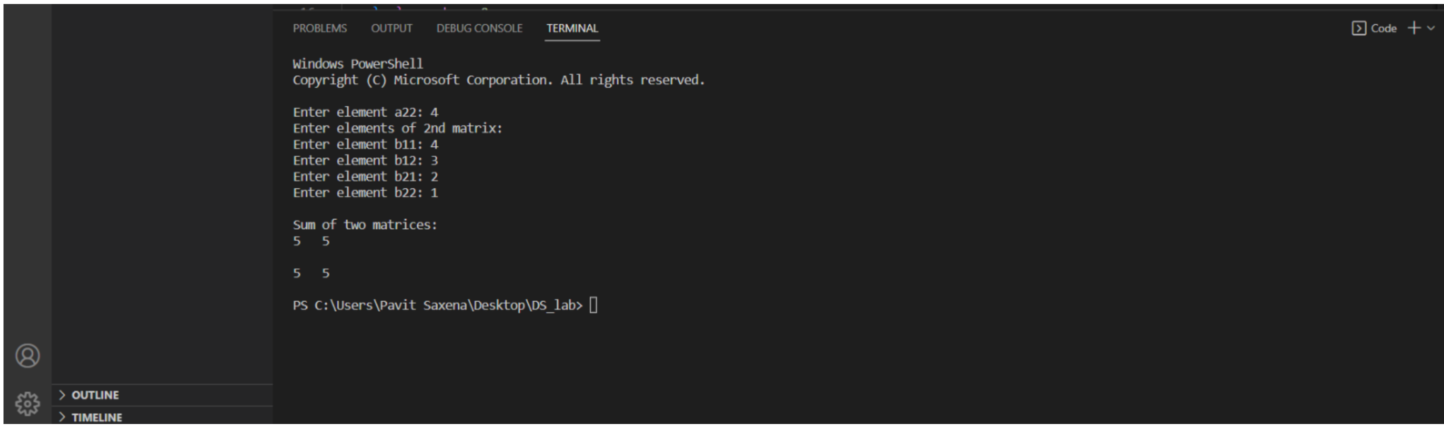
```
Step 1: Get the value of m and n as input from the user.
Step 2: Repeat step 3, for i = 1 to m.
Step 3: Repeat step 4, for j = 1 to n.
Step 4: Get the values of elements for matrix A as a[i] [j].
Step 5: Repeat step 6, for i = 1 to m.
Step 6: Repeat step 7, for j = 1 to n.
Step 7: Get the values of elements for matrix B as b[i] [j].
Step 8: Repeat step 9, for i = 1 to m.
Step 9: Repeat step 10, for j = 1 to n.
Step 10: Calculate and print sum of matrices as s[i] [j] = a[i] [j] + b[i] [j].
```

CODE:

```
#include <stdio.h>

int main()

{    int r, c, a[100][100], b[100][100], sum[100][100], i, j;

printf("Enter the number of rows (between 1 and 100): ");

scanf("%d", &r);    printf("Enter the number of columns

(between 1 and 100): ");   scanf("%d", &c);    printf("\nEnter

elements of 1st matrix:\n");    for (i = 0; i < r; ++i)       for (j =

0; j < c; ++j) {        printf("Enter element a%d%d: ", i + 1, j +

1);

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

```
  }
  printf("Enter elements of 2nd matrix:\n");
for (i = 0; i < r; ++i)  {    for (j = 0; j < c; ++j)
{        printf("Enter element b%d%d: ", i + 1, j
+ 1);       scanf("%d", &b[i][j]);
    }
}
  // adding two matrices
for (i = 0; i < r; ++i) {
for (j = 0; j < c; ++j) {
sum[i][j] = a[i][j] + b[i][j];
    }
}
  // printing the result    printf("\nSum
of two matrices: \n");   for (i = 0; i <
r; ++i)   {    for (j = 0; j < c; ++j) {
printf("%d   ", sum[i][j]);       if (j ==
c - 1) {         printf("\n\n");
      }
    }  }
return 0;
}
```

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Enter element a22: 4
Enter elements of 2nd matrix:
Enter element b11: 4
Enter element b12: 3
Enter element b21: 2
Enter element b22: 1

Sum of two matrices:
5   5

5   5

PS C:\Users\Pavit Saxena\Desktop\DS_lab> 
```

Multiplication of Two Arrays

To multiply two matrixes sufficient and necessary condition is "number of columns in matrix A = number of rows in matrix B". Loop for each row in matrix A.
Loop for each columns in matrix B and initialize output matrix C to 0. This loop will run for each rows of matrix A.
Loop for each columns in matrix A.
Multiply A[i,k] to B[k,j] and add this value to C[i,j]
Return output matrix C.Algorithm:

Algorithm:

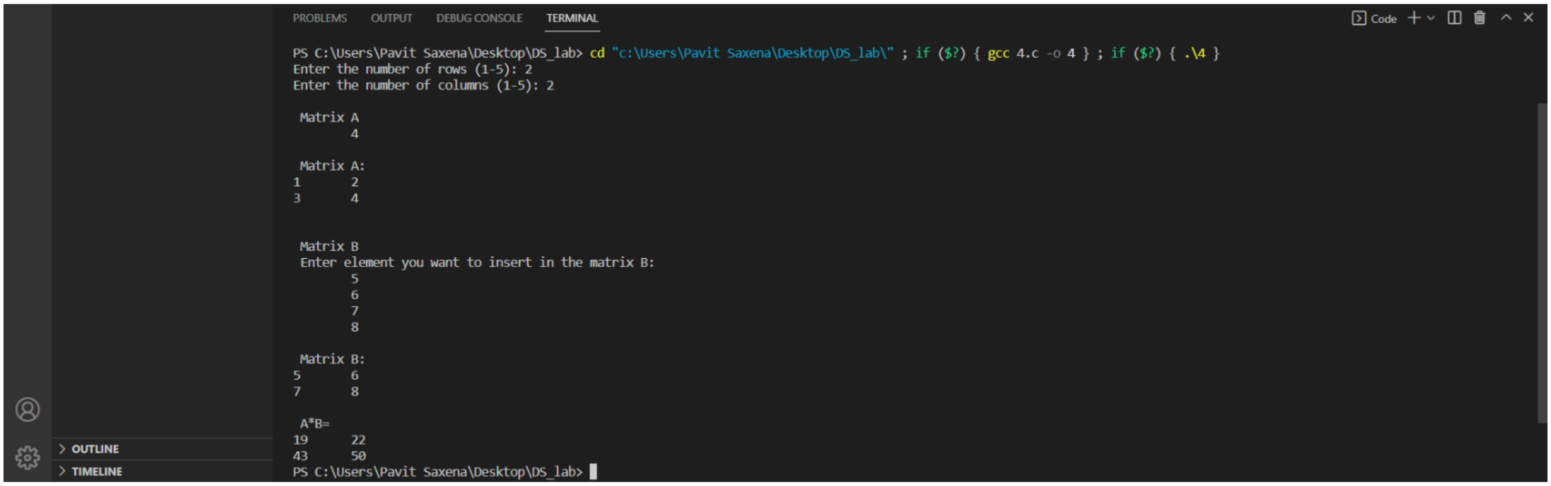
```
Matrix-Multiply(A, B)
if columns [A] ≠ rows [B]
then error "incompatible dimensions"
else
for i =1 to rows [A]
for j = 1 to columns [B]
C[i, j] =0
for k = 1 to columns [A]
C[i, j]=C[i, j]+A[i, k]*B[k, j]
return C
```

CODE:

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

```
int main()
{
    int C[10][10], A[10][10], B[10][10], i, j, l, row, col;
    printf("Enter the number of rows (1-5): ");
    scanf("%d", &row);
    printf("Enter the number of columns (1-5): ");
    scanf("%d", &col);
    printf("\n Matrix A\n");
    printf("\n Enter element you want to insert in the matrix A:\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            printf("\t");
            scanf("%d", &A[i][j]);
        }
    }
    printf("\n Matrix A:\n");
    for (i = 0; i < row; i++)

    {

        for (j = 0; j < col; j++)

        {

            printf("%d", A[i][j]);

            printf("\t");

}

        printf("\n");

    }

    printf("\n\n Matrix B");

    printf("\n Enter element you want to insert in the matrix B:\n");

    for (i = 0; i < row; i++)

    {

        for (j = 0; j < col; j++)

        {
            printf("\t");

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

        }
    }
    printf("\n Matrix B:\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            printf("%d", B[i][j]);
            printf("\t");
}
        printf("\n");
    }
    printf("\n A*B=\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < row; j++)
        {
C[i][j] = 0;
            for (l = 0; l < col; l++)
            {

                C[i][j] = C[i][j] + A[i][l] * B[l][j];

            }

            printf("%d", C[i][j]);

            printf("\t");

}

        printf("\n");

    }

    return 0;

 printf("Made by PAVIT SAXENA ");

}
```

OUTPUT:



```
PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Desktop\DS_lab\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
Enter the number of rows (1-5): 2
Enter the number of columns (1-5): 2

 Matrix A
        4

 Matrix A:
1       2
3       4


 Matrix B
 Enter element you want to insert in the matrix B:
        5
        6
        7
        8

 Matrix B:
5       6
7       8

 A*B=
19      22
43      50
PS C:\Users\Pavit Saxena\Desktop\DS_lab>
```

Program:3

Write a program in C to TRANSPOSE OF A 2D ARRAY.

Algorithm:

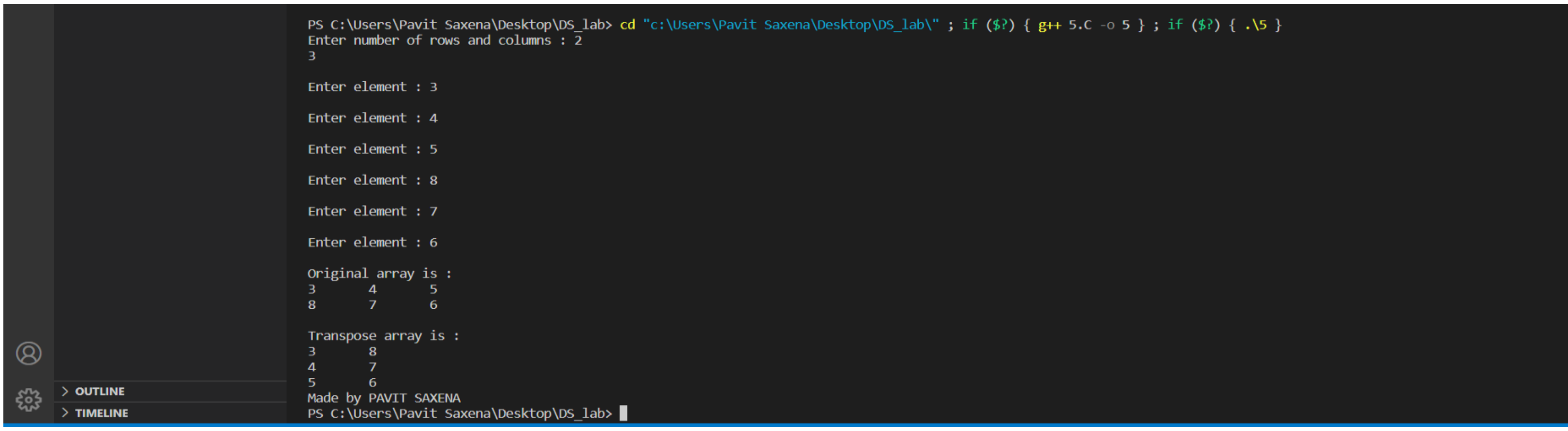
```
1.  Set i=0 and Repeat step 2 to 5 while i< 2
2.  Set j=0 and Repeat step 3 and 4 while j<2
3.  Input the values of A[i][j] 4. Set j=j+1
    [End of step 2 loop]
5.  Set i=i+1
    [End of step 1 loop]
6.  Set i=0 and Repeat step 7 to 10 while i< 2
7.  Set j=0 and Repeat step 8 and 9 while j<2
8.  Set B[j][i]=A[i][j] 9. Set j=j+1
    [End of step 6 loop]
10. Set i=i+1
    [End of step 7 loop]
11. Set i=0 and Repeat step 12 to 15 while i< 2
12. Set j=0 and Repeat step 13 and 14 while j<2
13. Write: the value of B[i][j] 14. Set j=j+1
    [End of step 11 loop]
15. Set i=i+1 [End of
step 12 loop]
16. Exit.
```

CODE:

```
#include<stdio.h>

void main()

{     int

c,r,i,j;

   printf("Enter number of rows and columns : ");

scanf("%d %d",&r,&c);

   int arr[r][c];

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

for(j=0;j<c;j++)

      {           printf("\nEnter

element : ");

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

      }

   }
   printf("\nOriginal array is : \n");

for(i=0;i<r;i++)

   {

for(j=0;j<c;j++)

      {

printf("%d\t",arr[i][j]);

      }

printf("\n");

   }

   printf("\nTranspose array is : \n");

for(i=0;i<c;i++)

   {

for(j=0;j<r;j++)
```

```
        {
printf("%d\t",arr[j][i]);
        }
printf("\n");
    }
printf("Made by PAVIT SAXENA ");
}
```

OUTPUT:



```
PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Desktop\DS_lab\" ; if ($?) { g++ 5.C -o 5 } ; if ($?) { .\5 }
Enter number of rows and columns : 2
3

Enter element : 3

Enter element : 4

Enter element : 5

Enter element : 8

Enter element : 7

Enter element : 6

Original array is :
3       4       5
8       7       6

Transpose array is :
3       8
4       7
5       6
Made by PAVIT SAXENA
PS C:\Users\Pavit Saxena\Desktop\DS_lab> █
```

Program:4 (16)

Write a program in C to implement LINEAR SEARCH.

Algorithm : Linear_search (A, N, X)

Here A is an array with N elements. X is the element to be searched.

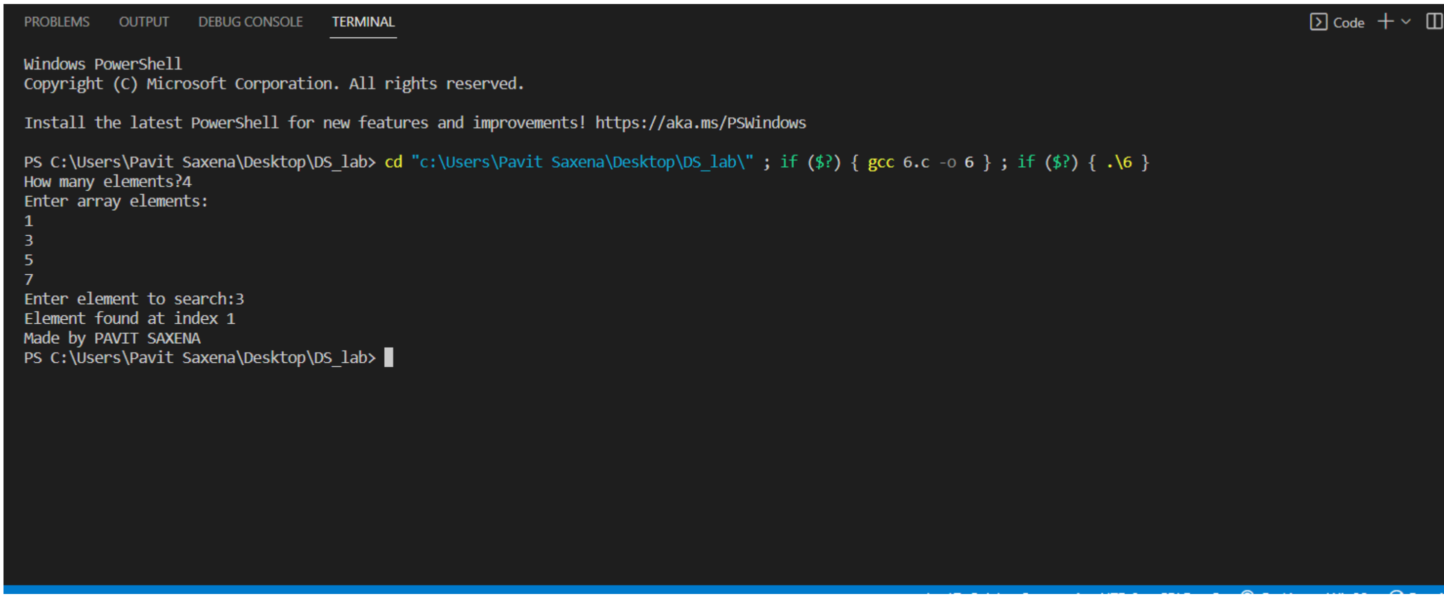
```
Step1: [Search the array]
    Repeat for i = 1 to N
        If A[i] = X then
            Return (i) and exit
Step 2:[Element not found]
```

```
            Return(0)
            Exit.
```

CODE:

```
#include<stdio.h>


int main()

{      int a[20],i,x,n;

printf("How many elements?");

scanf("%d",&n);


    printf("Enter array elements:\n");

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


    printf("Enter element to search:");
scanf("%d",&x);


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

if(a[i]==x)             break;

          if(i<n)

   printf("Element found at index %d\n",i);

else

   printf("Element not found\n");

printf("Made by PAVIT SAXENA");

return 0;

}
```

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Desktop\DS_lab\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
How many elements?4
Enter array elements:
1
3
5
7
Enter element to search:3
Element found at index 1
Made by PAVIT SAXENA
PS C:\Users\Pavit Saxena\Desktop\DS_lab> 
```

Program:5(17)

Write a program in C to implement BINARY SEARCH

Algorithm :

Note: Binary search can only be applied on sorted list of elements.

```
Binary_search(A, N, X)

Here A is an array with N elements. X is the element to be searched.

Step1: [Initialise variables]

                    First := 1

          Last := N

Middle :=Int((First +Last)/2)

Step 2: Repeat step 3 and 4 while first< = Last and A[Middle] <> X

Step 3: If X <A[Middle] then

                             Last := Middle-1

                    Else
                             First := Middle+1
```

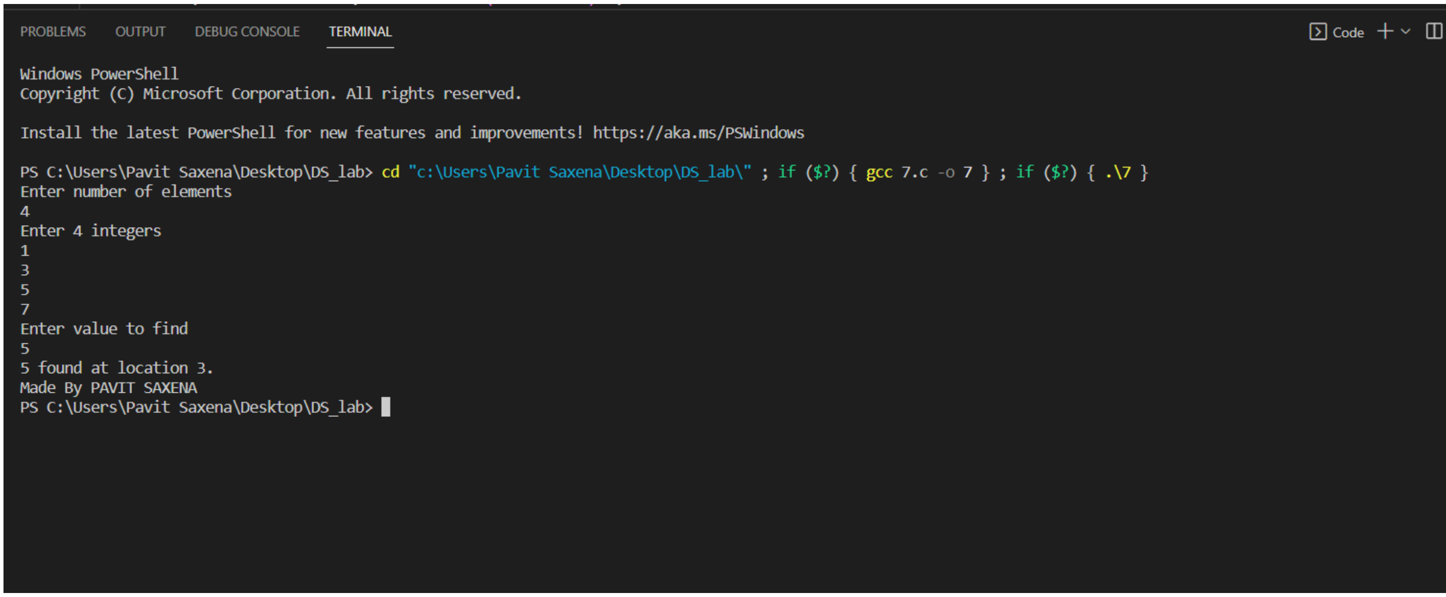
```
Step 4:Middle = int((First + Last)/2)

Step 5: If A[Middle] = X then

                      return(Middle)

 Else                 return(0)

Step 6: Exit.
```

CODE:

```
 #include <stdio.h>   int main()  {    int c, first,

last, middle, n, search, array[100];


  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]);


  printf("Enter value to find\n");

scanf("%d", &search);

   first = 0;

last = n - 1;

middle =

(first+last)/2;


  while (first <= last) {      if (array[middle] < search)

first = middle + 1;      else if (array[middle] == search) {
```

```
printf("%d found at location %d.\n", search, middle+1);
break;
    }      else
last = middle - 1;


    middle = (first + last)/2;
  }    if (first > last)      printf("Not found! %d isn't present
in the list.\n", search);
printf("Made By PAVIT SAXENA");


  return 0;
}
```

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Desktop\DS_lab\" ; if ($?) { gcc 7.c -o 7 } ; if ($?) { .\7 }
Enter number of elements
4
Enter 4 integers
1
3
5
7
Enter value to find
5
5 found at location 3.
Made By PAVIT SAXENA
PS C:\Users\Pavit Saxena\Desktop\DS_lab> 
```

Program:6(18)

Write a program in C to implement BUBBLE SORTING.

Algorithm:

Buuble_sort (A, N)

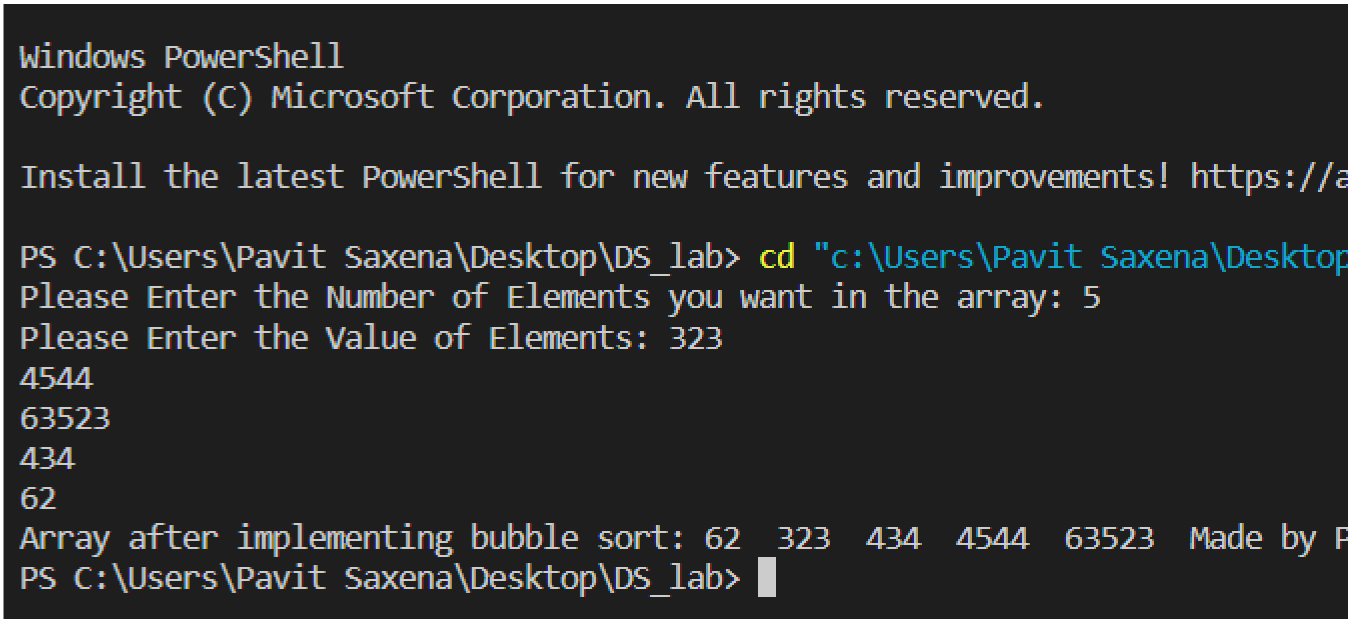
```
 Let A be a linear array with N elements
 Step 1: Repaet steps 2 and 3 for i =1 to N-1
 Step 2: Repeat step 3 for j= 1 to N-1
Step 3: If a[j] >A[j+1] then Temp := A[j] A[j] := A[j+1] A[j+1] := Temp
 Step 4: Exit
```

CODE:

```
#include <stdio.h>
int main()
{
  int arr[50], num, x, y, temp;
   printf("Please Enter the Number of Elements you want in the array: ");
    scanf("%d", &num);
   printf("Please Enter the Value of Elements: ");
    for(x = 0; x < num; x++)
   scanf("%d", &arr[x]);
    for(x = 0; x < num - 1; x++)
    {
      for(y = 0; y < num - x - 1; y++)
      {
         if(arr[y] > arr[y + 1])
         {
           temp = arr[y];
           arr[y] = arr[y + 1];
           arr[y + 1] = temp;
         }
      }
    }
         printf("Array after implementing bubble sort: ");
```

```
        for(x = 0; x < num; x++)

        {

        printf("%d  ", arr[x]);

        }

  printf("Made by PAVIT SAXENA");

   return 0;

}
```

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://a

PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Desktop
Please Enter the Number of Elements you want in the array: 5
Please Enter the Value of Elements: 323
4544
63523
434
62
Array after implementing bubble sort: 62  323  434  4544  63523  Made by P
PS C:\Users\Pavit Saxena\Desktop\DS_lab>
```

Program:7(19)

Write a program in C to implement selection sort.

Algorithm:

Selection_sort (A, N) Let A be a linear array with N elements.

Step 1: Repeat steps 2 to 4 for k = 1 to N-1

Step 2: set MIN := A[k] Position := k

Step 3: [Make a pass and obtain the element with smallest value] Repeat for i = k+1 to N If MIN > A[i] then MIN := A[i] and Position := i

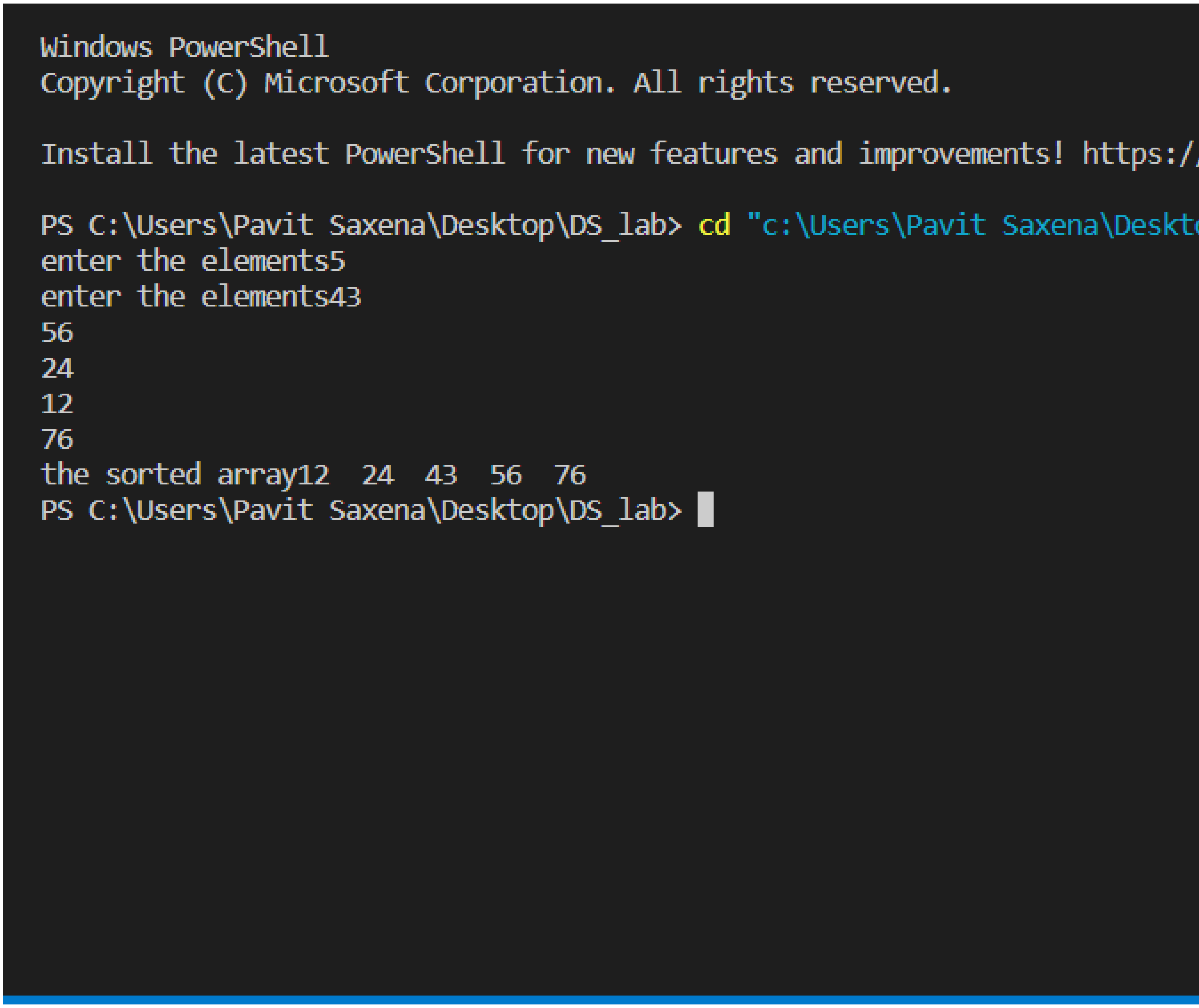
Step 4: [Exchange elements] If Position<> k then Temp := A[k] A[k] := A[Position] A[Position] := Temp

5: Exit

CODE:

```
#include<stdio.h> void main() {     int
i,j,n,a[10],temp,min,p;     printf("enter the elements");
scanf("%d",&n);
    printf("enter the elements");
for(i=0;i<n;i++){
scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {         min =i;
for(j=i+1;j<n;j++)
        {
if(a[j]<a[min])
min= j;
        }
        temp= a[min];
a[min]= a[i];
a[i]= temp;
    }          printf("the  sorted
array");      for(i=0;i<n;i++)
printf("%d  ",a[i]);


}
```

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https:/

PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Deskt
enter the elements5
enter the elements43
56
24
12
76
the sorted array12  24  43  56  76
PS C:\Users\Pavit Saxena\Desktop\DS_lab> 
```

Program:8(20)

Write a program in C to implement INSERTION SORT.

Algorithm:

Consider an array with N elements. Step 1: A[1] by itself is sorted

Step 2: A[2] is inserted before or after A[1], so that A[1] and A[2] are sorted.

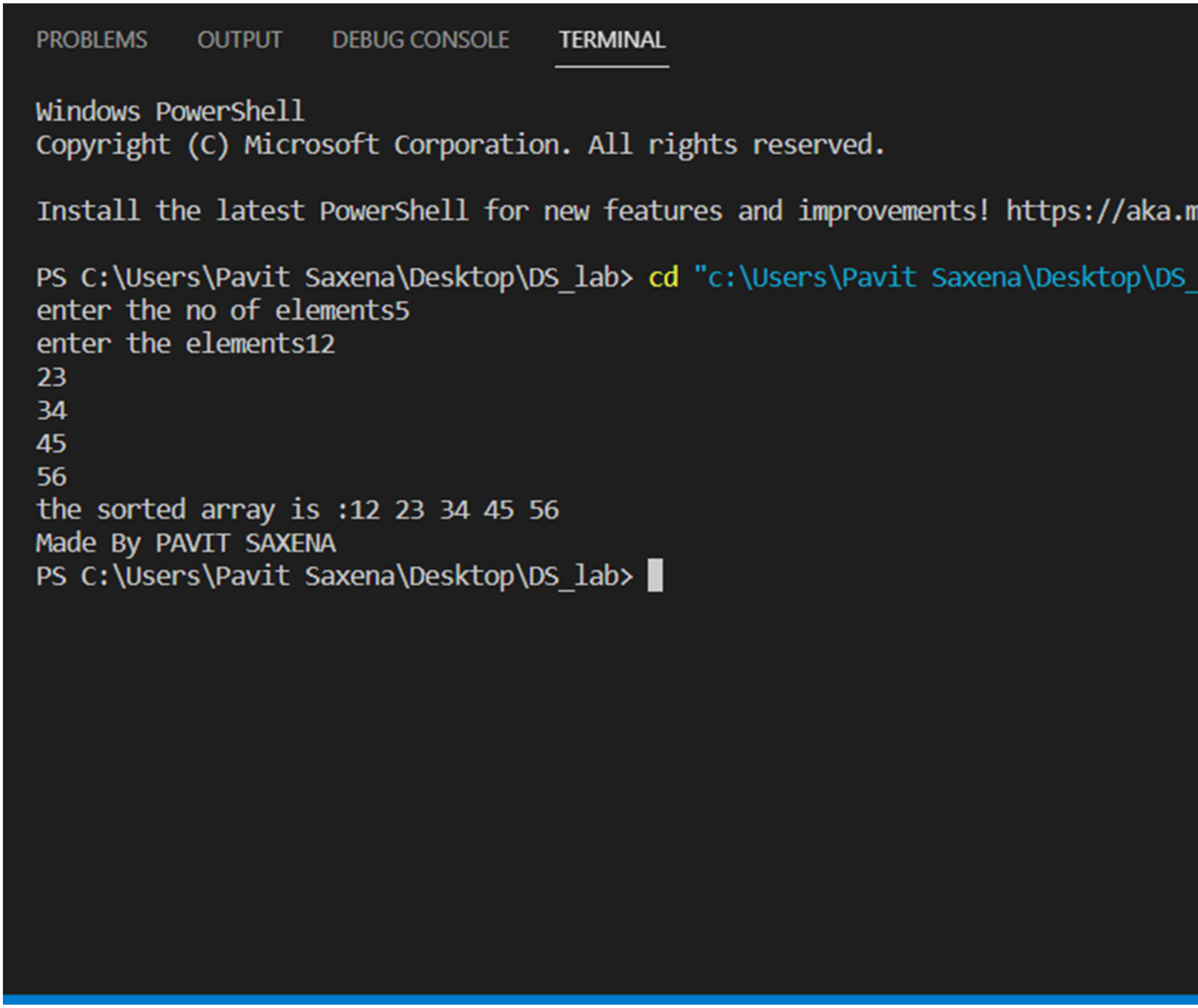
Step 3: Similarly A[3] is inserted so that A[1], A[2] and A[3] are sorted.

Step 4: This process is continued till all the elements are sorted

CODE:

```
#include<stdio.h> void main() {

int a[10],i,j,n,sto,u,min;
printf("enter the no of elements");
scanf("%d",&n);
    printf("enter the elements");
for(i=0;i<n;i++)
    {
    scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++)
    {
j=i;
        while(j>0&&a[j-1]>a[j])
        {            sto
= a[j];            a[j]=
a[j-1];            a[j-
1]= sto;             j--;
}
    }     printf("the sorted
array");     for(i=0;i<n;i++)
printf("%d",a[i]);
printf("Made  by PAVIT SAXENA");


}
```

OUTPUT:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.m

PS C:\Users\Pavit Saxena\Desktop\DS_lab> cd "c:\Users\Pavit Saxena\Desktop\DS_
enter the no of elements5
enter the elements12
23
34
45
56
the sorted array is :12 23 34 45 56
Made By PAVIT SAXENA
PS C:\Users\Pavit Saxena\Desktop\DS_lab> 
```

Write a program in C to implement MERGE SORTING

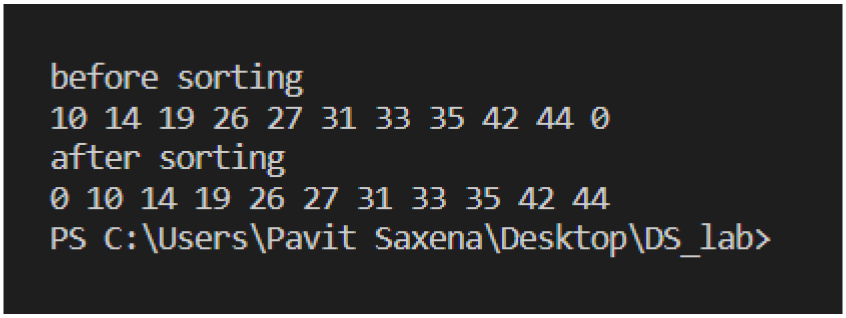
CODE:

```
#include <stdio.h>
int b[10], a[11] = {10, 14, 19, 26, 27, 31, 33, 35, 42, 44, 0};
void merging(int low, int mid, int high)
{
   int l1, l2, i;

   for (l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++)
```

```
    {
        if (a[l1] <= a[l2])
            b[i] = a[l1++];
        else
            b[i] = a[l2++];
    }

    while (l1 <= mid)
        b[i++] = a[l1++];

    while (l2 <= high)
        b[i++] = a[l2++];

    for (i = low; i <= high; i++)
        a[i] = b[i];
}
void sort(int low, int high)
{
    int mid;

    if (low < high)
    {
        mid = (low + high) / 2;
        sort(low, mid);
        sort(mid + 1, high);
        merging(low, mid, high);
    }
    else
    {
        Return 0;
    }
```

```
}

int main()
{
    int i;

    printf("before sorting\n");

    for (i = 0; i <= 10; i++)
        printf("%d ", a[i]);

    sort(0, 10);

    printf("after sorting\n");

    for (i = 0; i <= 10; i++)
        printf("%d ", a[i]);
}
```

OUTPUT:



```
before sorting
10 14 19 26 27 31 33 35 42 44 0
after sorting
0 10 14 19 26 27 31 33 35 42 44
PS C:\Users\Pavit Saxena\Desktop\DS_lab>
```

23. Write a program in C to implement HEAP SORTING

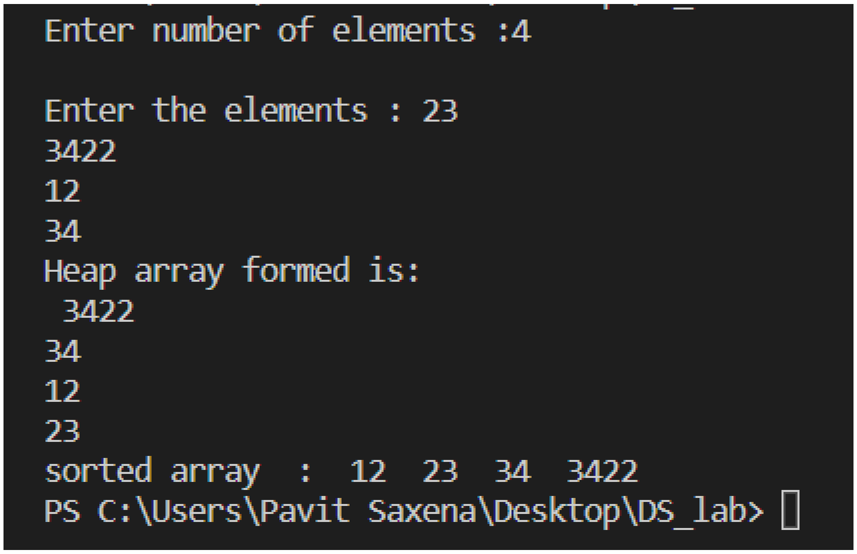
CODE:

```
#include <stdio.h>
int main()
{
int h[20],num,i,j,root,t,x;
```

```
printf("Enter number of elements :");
scanf("%d", &num);
printf("\nEnter the elements : ");
for (i = 0; i < num; i++)
scanf("%d", &h[i]);
for(i=0;i<num;i++)
{
x=i;
do
{
root = (x - 1) / 2;
if (h[root] < h[x])
{
t = h[root];
h[root] = h[x];
h[x] = t;
}
x = root;
} while (x != 0);
}
printf("Heap array formed is: ");
for (i = 0; i < num; i++)
printf("%d", h[i]);
for (j = num - 1; j >= 0; j--)
{
t = h[0];
h[0] = h[j];
h[j] = t;
root = 0;
do
{
```

```
x = 2 * root + 1;
if ((h[x] < h[x + 1]) && x < j-1)
x++;
if (h[root]<h[x] && x<j)
{
t = h[root];
h[root] = h[x];
h[x] = t;
}
root = x;
} while (x < j);
}
printf("sorted array  : ");
for (i = 0; i < num; i++)
printf(" %d ", h[i]);
}
```

OUTPUT:



```
Enter number of elements :4

Enter the elements : 23
3422
12
34
Heap array formed is:
 3422
34
12
23
sorted array  :  12  23  34  3422
PS C:\Users\Pavit Saxena\Desktop\DS_lab> 
```