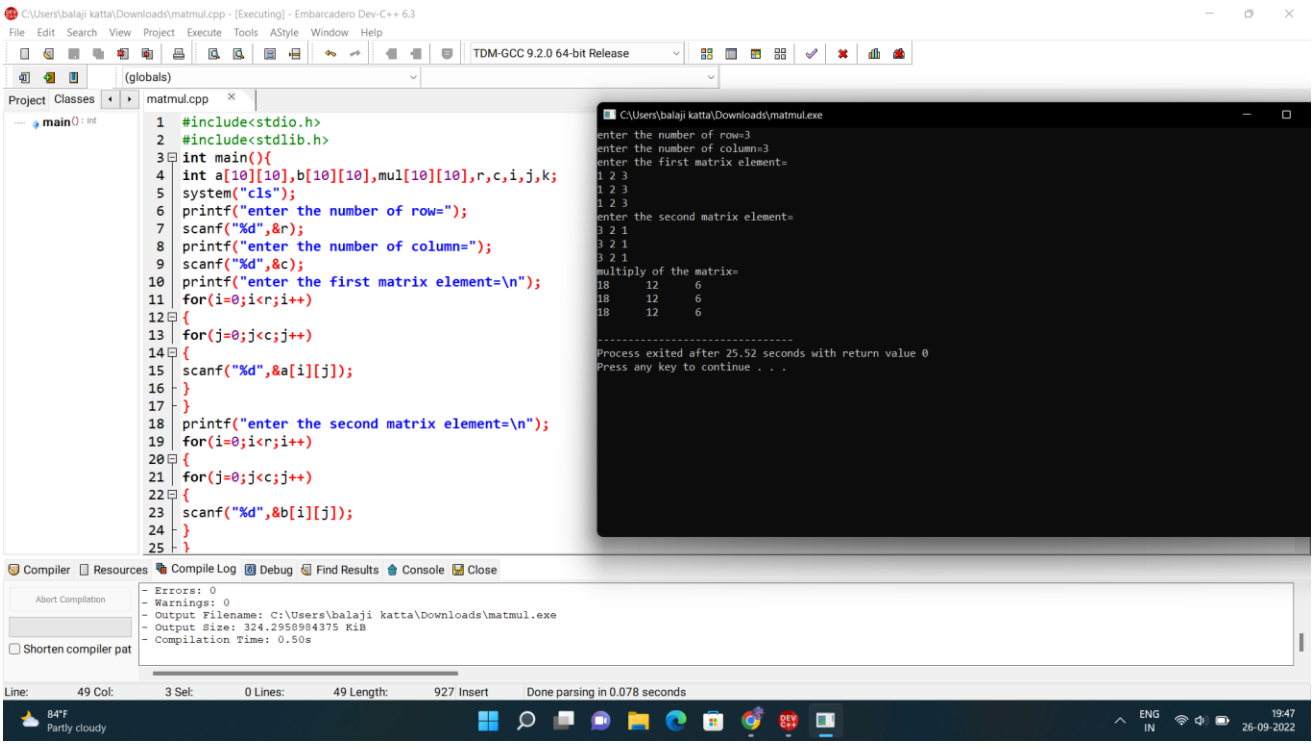
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
COMPUTER SCIENCE AND ENGINEERING PROGRAMME

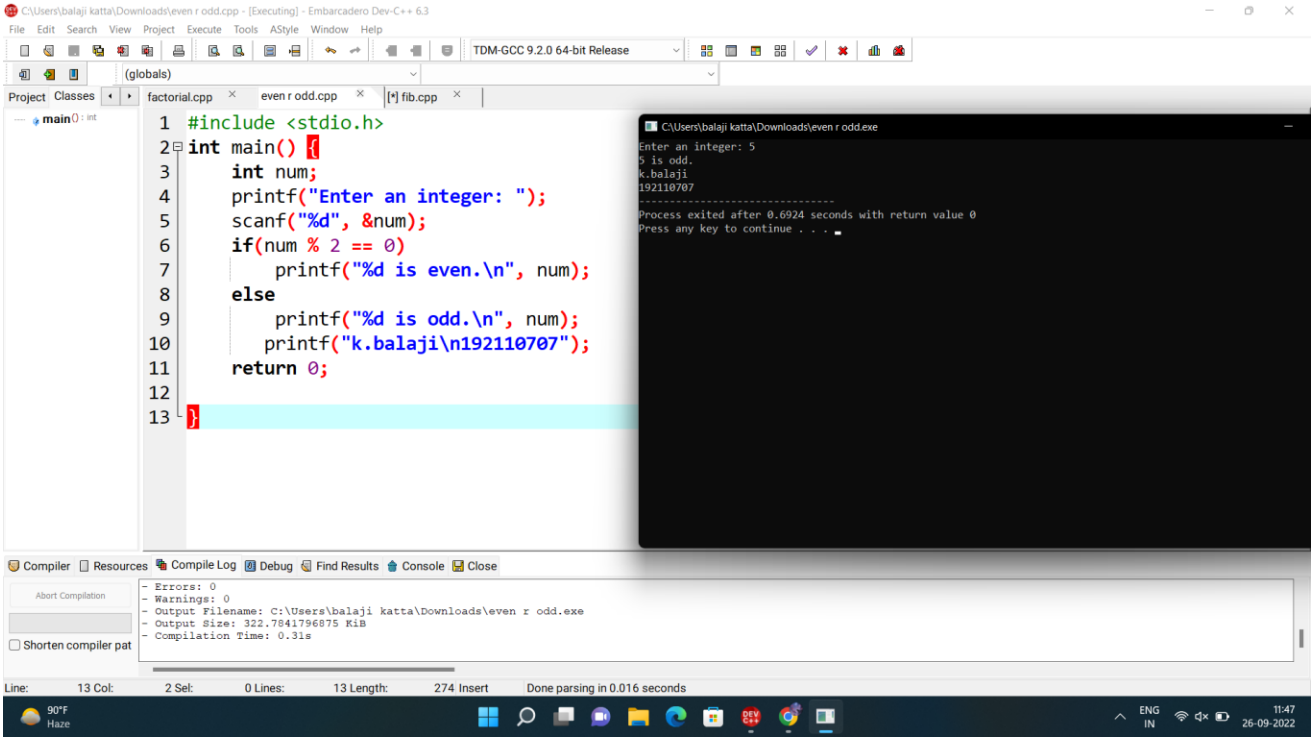
CSA03 - DATA STRUCTURES

LIST OF PROGRAMS

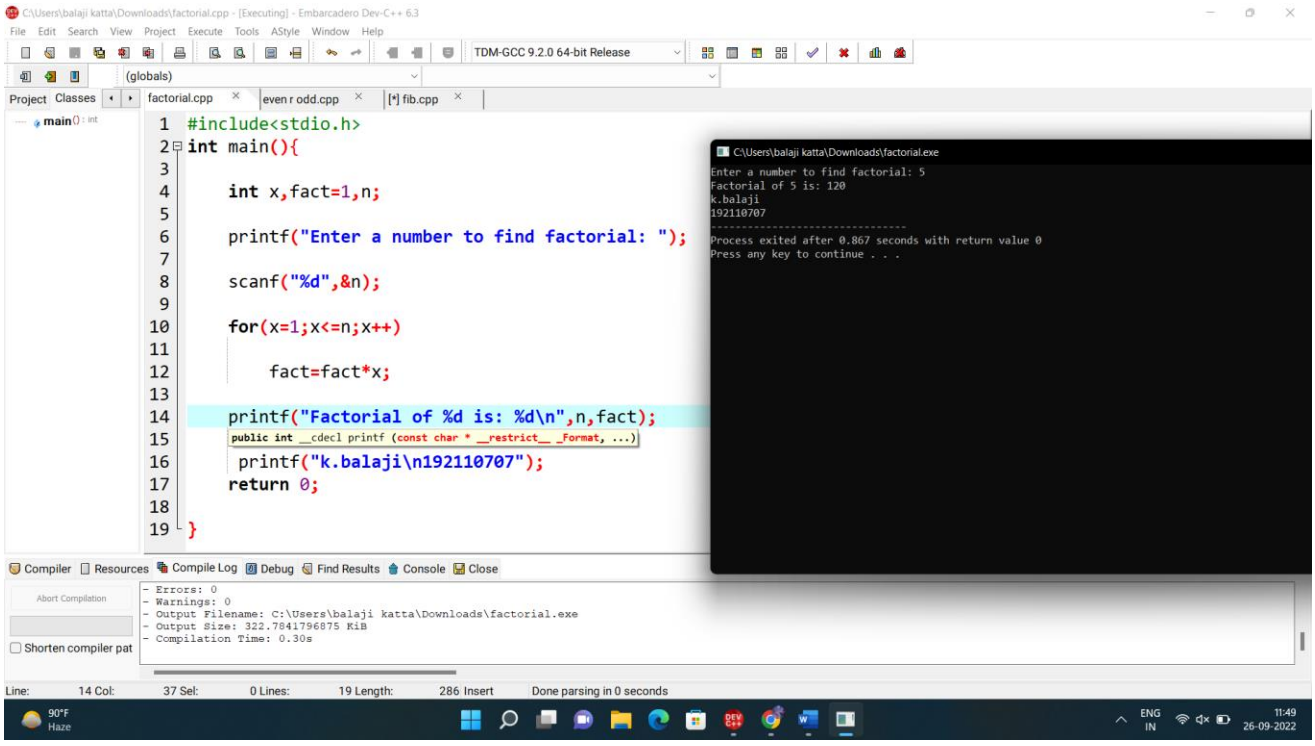
1. Write a C program to perform Matrix Multiplication



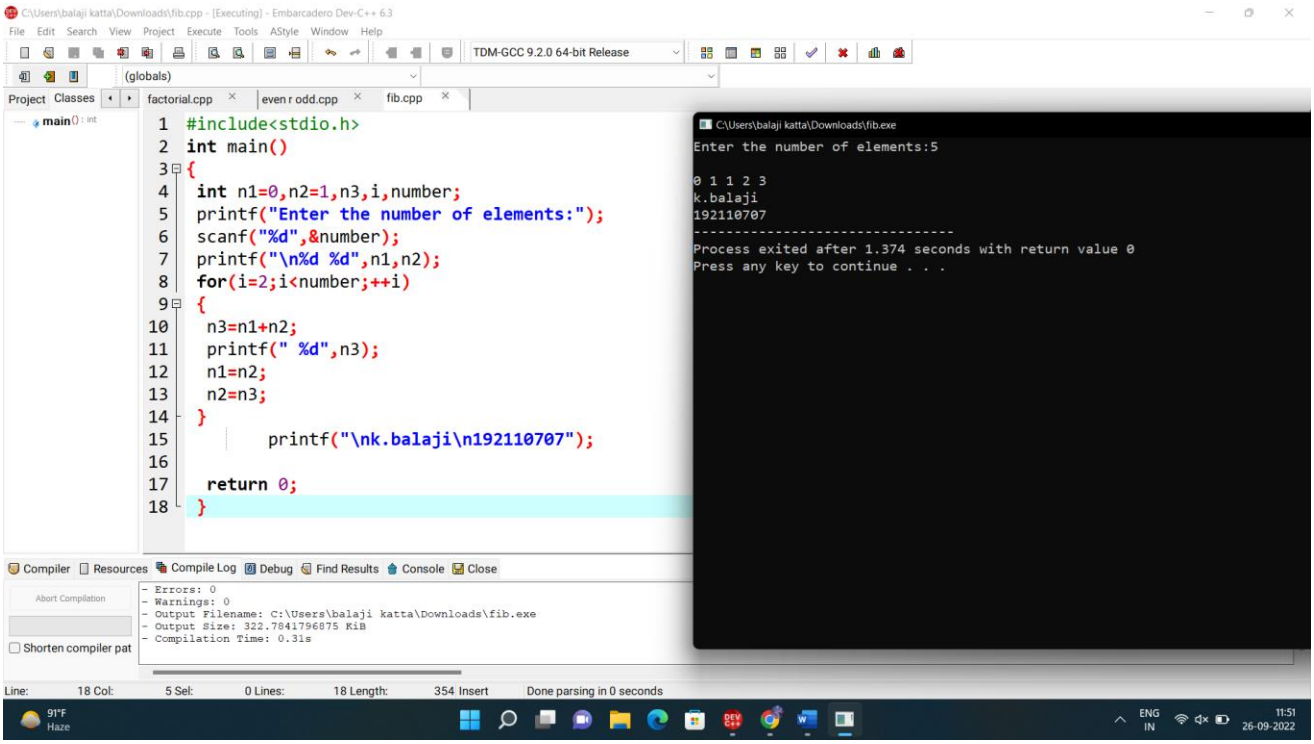
2. Write a C program to find Odd or Even number from a given set of numbers



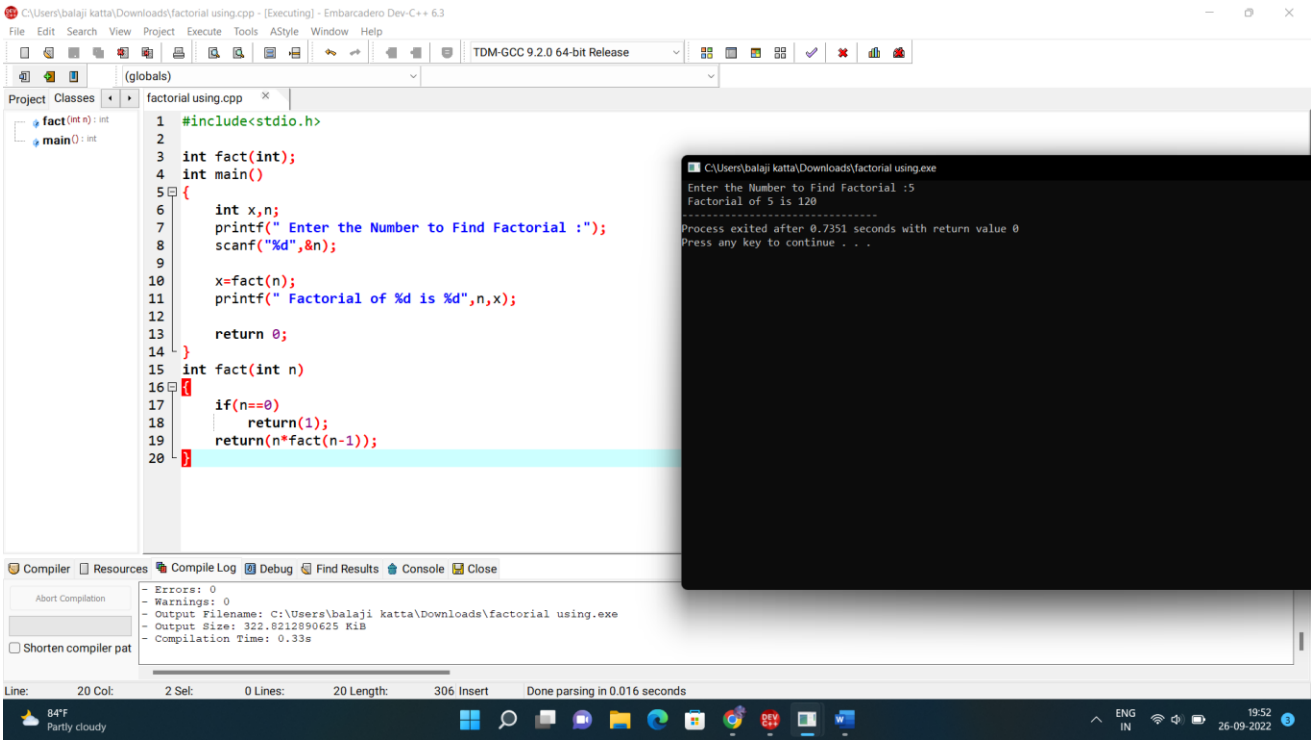
3. Write a C program to findFactorial of a given number without using Rec ursion



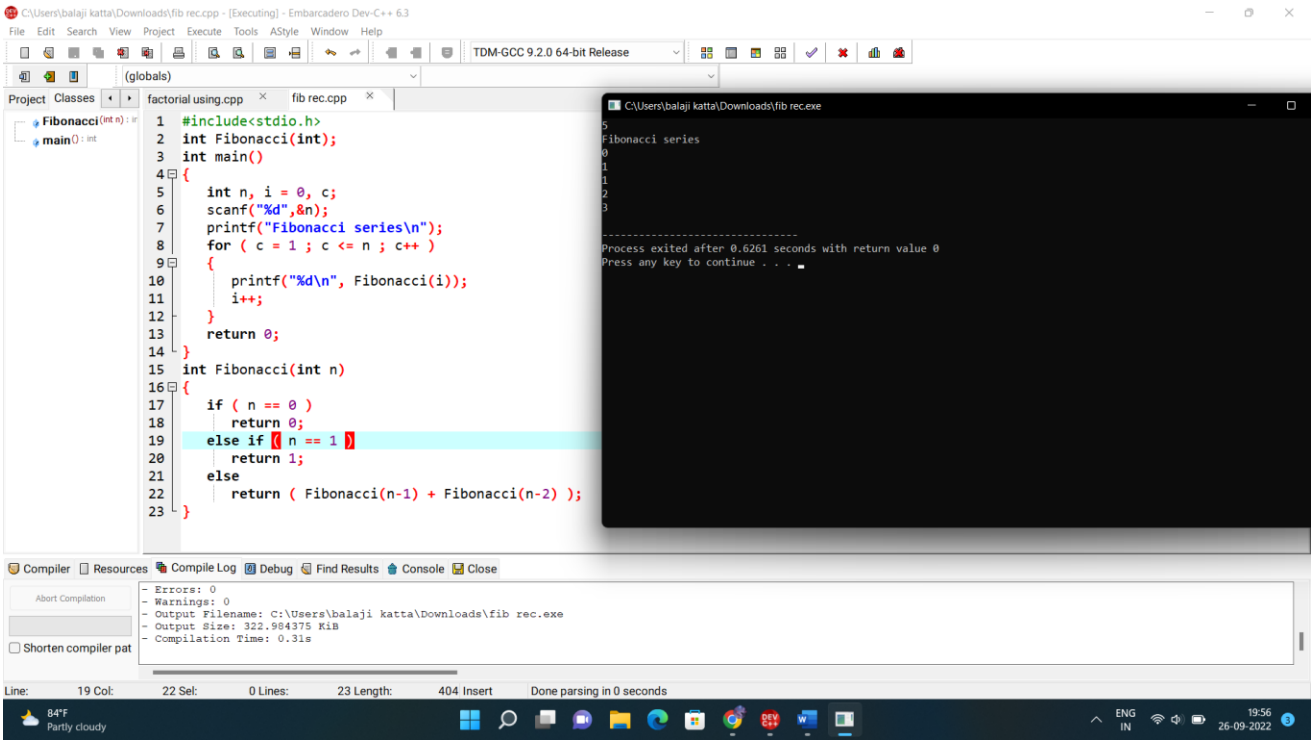
4. Write a C program to find Fibonacci series without using Recursion



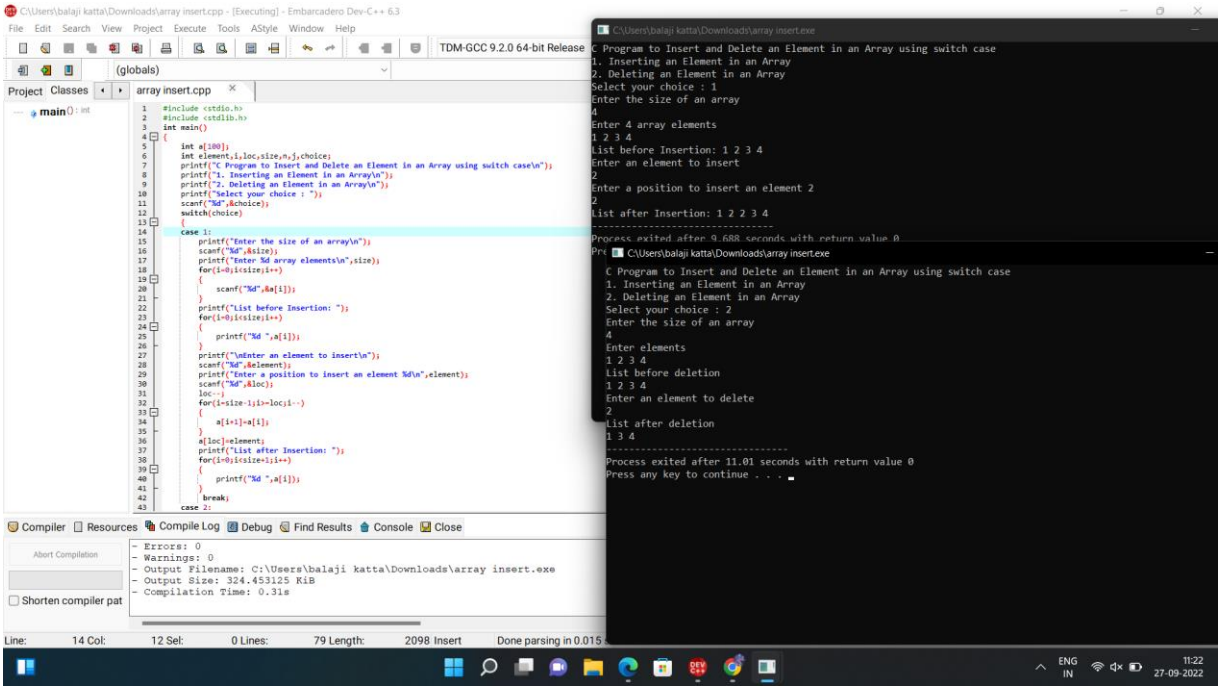
5. Write a C program to find Factorial of a given number using Recursion



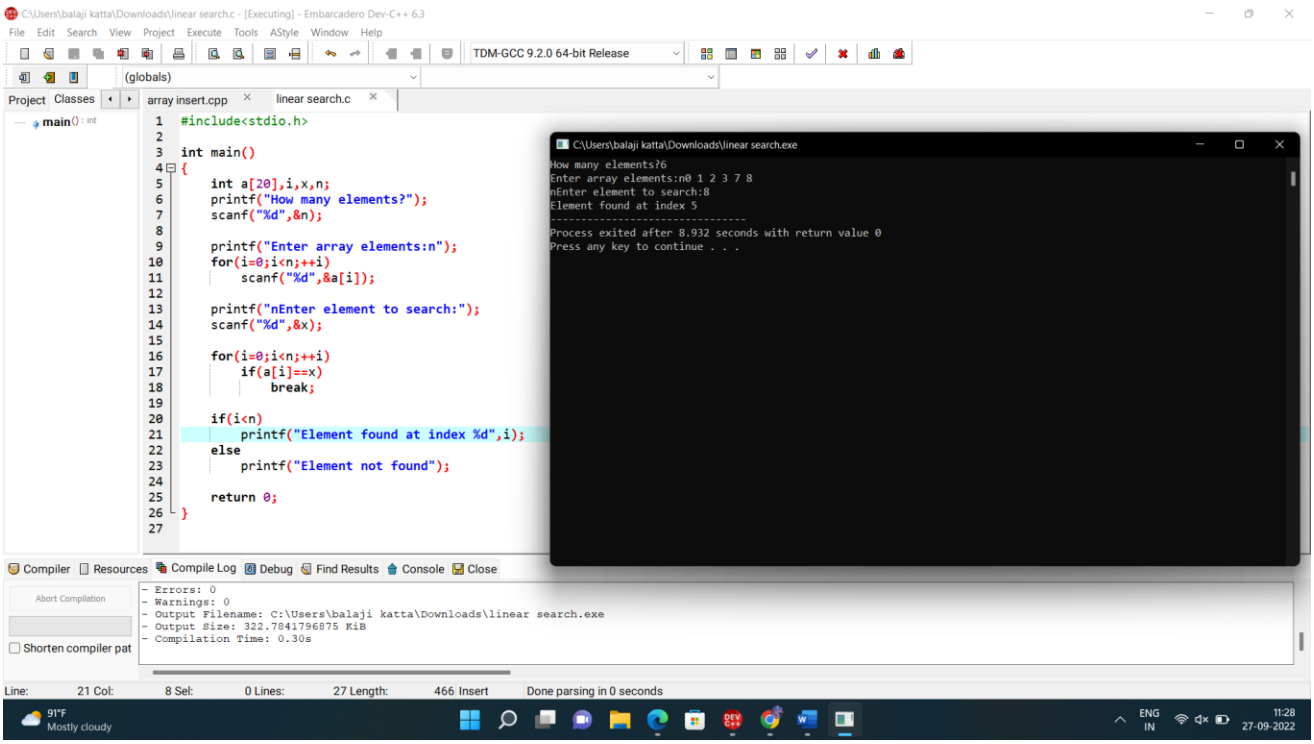
6. Write a C program to find Fibonacci series using Recursion



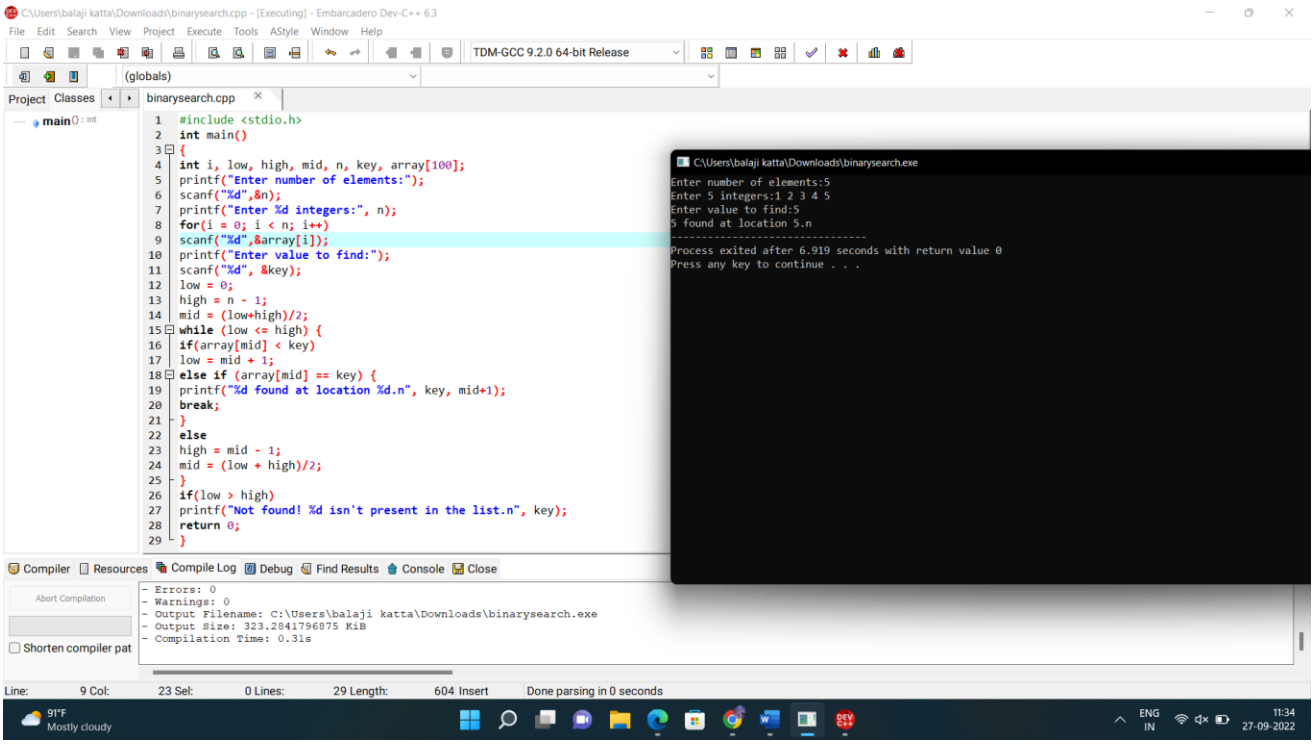
7. Write a C program to implement Array operations such as Insert, Delete and Display.



8. Write a C program to search a number using Linear Search method



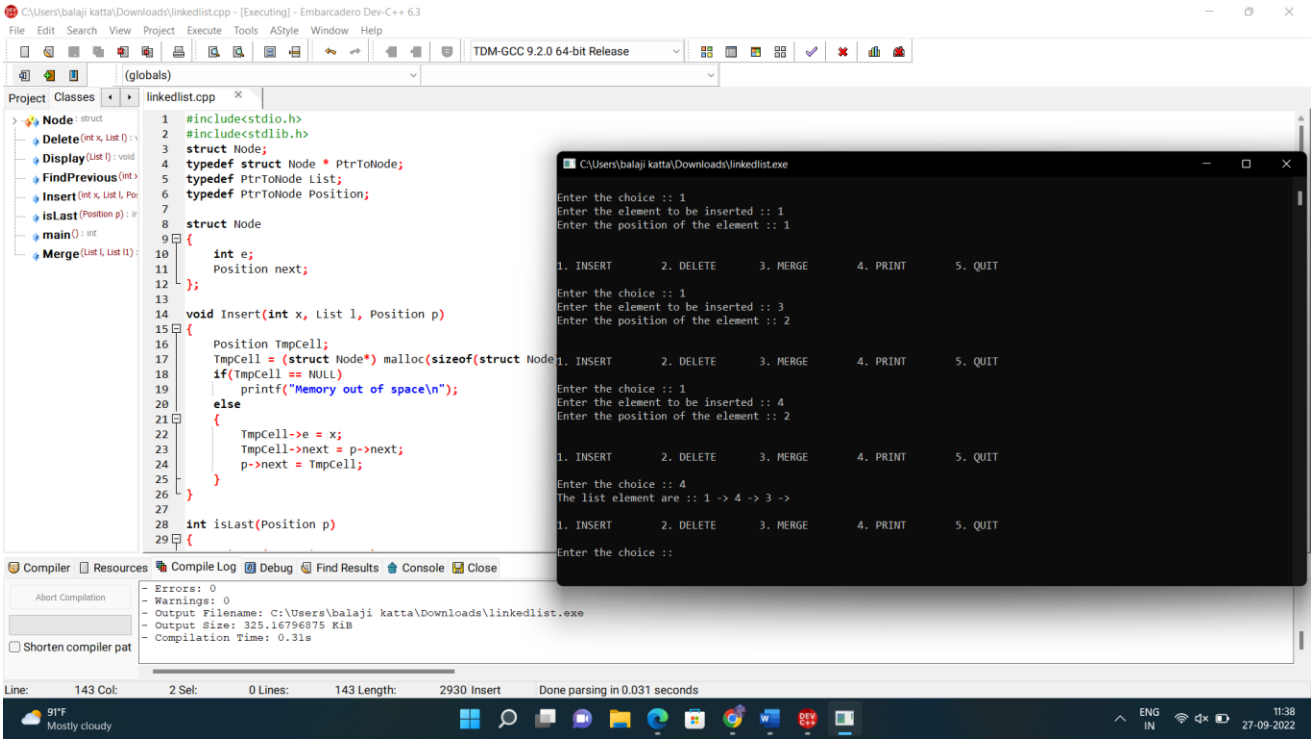
9. Write a C program to search a number using Binary Search method



```
#include <stdio.h>
int main()
{
int i, low, high, mid, n, key, array[100];
printf("Enter number of elements:");
scanf("%d",&n);
printf("Enter %d integers:", n);
for(i = 0; i < n; i++)
scanf("%d",&array[i]);
printf("Enter value to find:");
scanf("%d", &key);
low = 0;
high = n - 1;
mid = (low+high)/2;
while (low <= high) {
if(array[mid] < key)
low = mid + 1;
else if (array[mid] == key) {
printf("%d found at location %d.n", key, mid+1);
break;
}
else
high = mid - 1;
mid = (low + high)/2;
}
if(low > high)
printf("Not found! %d isn't present in the list.n", key);
return 0;
}
```

```
C:\Users\balaji katta\Downloads\binarysearch.exe
Enter number of elements:5
Enter 5 integers:1 2 3 4 5
Enter value to find:5
5 found at location 5.n

Process exited after 6.919 seconds with return value 0
Press any key to continue . . .
```

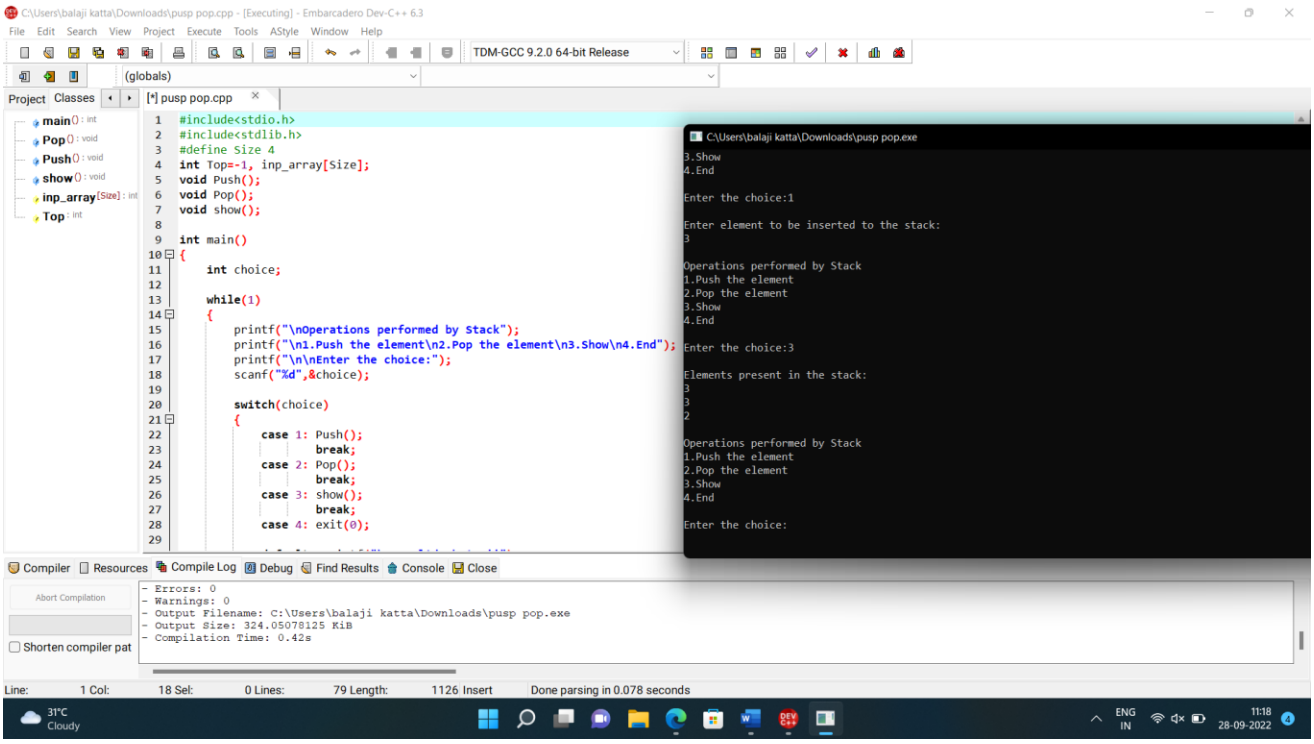
10. Write a C program to implement Linked list operations



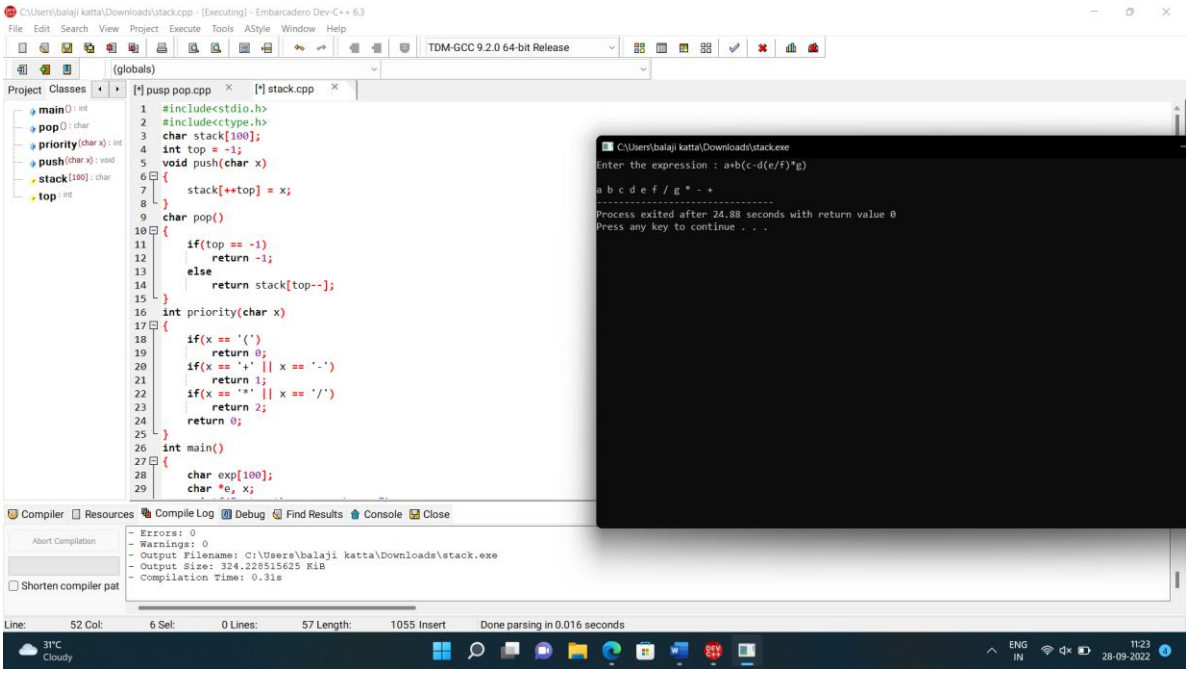
```
#include<stdio.h>
#include<stdlib.h>
struct Node;
typedef struct Node * PtrToNode;
typedef PtrToNode List;
typedef PtrToNode Position;

struct Node
{
    int e;
    Position next;
};

void Insert(int x, List l, Position p)
{
    Position TmpCell;
    TmpCell = (struct Node*) malloc(sizeof(struct Node
    if(TmpCell == NULL)
        printf("Memory out of space\n");
    else
    {
        TmpCell->e = x;
        TmpCell->next = p->next;
        p->next = TmpCell;
    }
}

int isLast(Position p)
{
```

```
C:\Users\balaji katta\Downloads\linkedlist.exe

Enter the choice :: 1
Enter the element to be inserted :: 1
Enter the position of the element :: 1

1. INSERT       2. DELETE       3. MERGE       4. PRINT       5. QUIT

Enter the choice :: 1
Enter the element to be inserted :: 3
Enter the position of the element :: 2

1. INSERT       2. DELETE       3. MERGE       4. PRINT       5. QUIT

Enter the choice :: 1
Enter the element to be inserted :: 4
Enter the position of the element :: 2

1. INSERT       2. DELETE       3. MERGE       4. PRINT       5. QUIT

Enter the choice :: 4
The list element are :: 1 -> 4 -> 3 ->

1. INSERT       2. DELETE       3. MERGE       4. PRINT       5. QUIT

Enter the choice ::
```

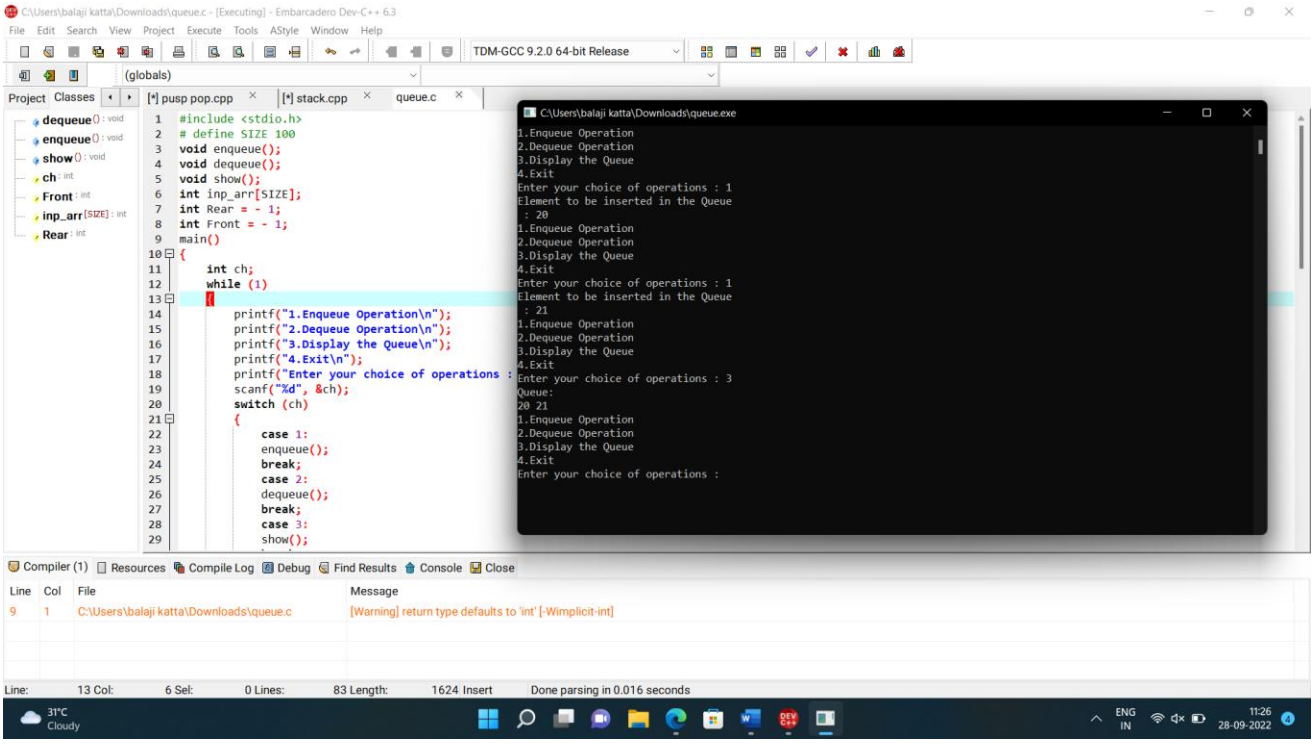
11. Write a C program to implement Stack operations such as PUSH, POP and PEEK



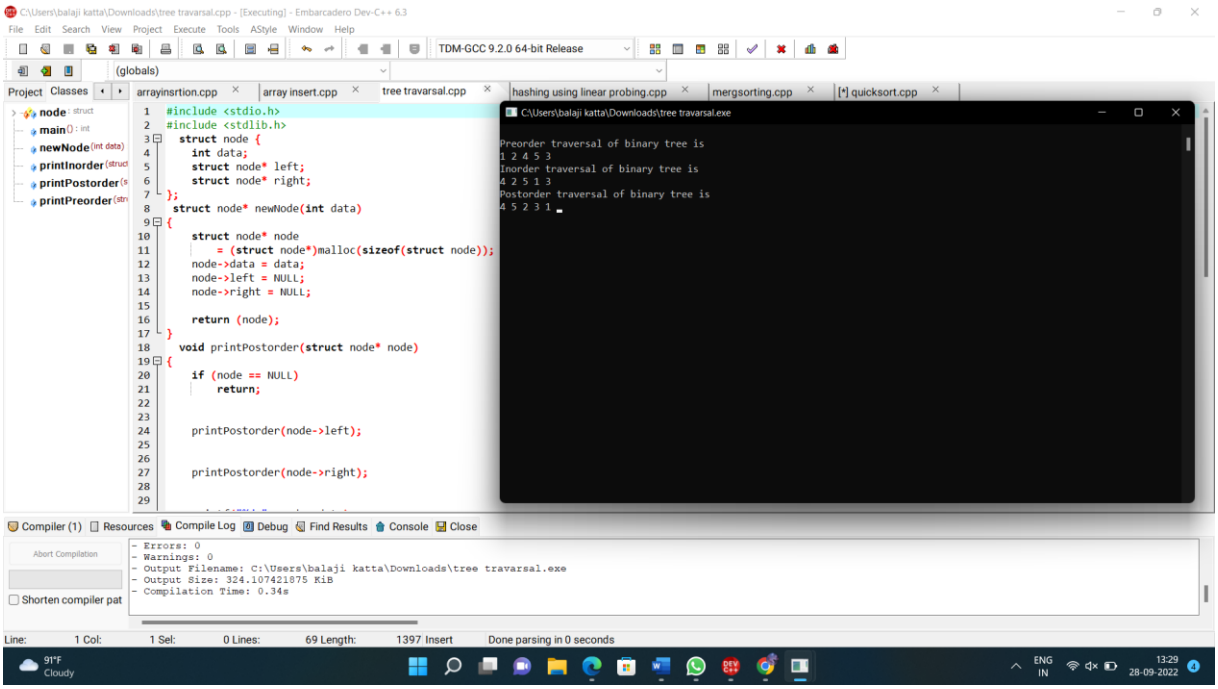
12. Write a C program to implement the application of Stack (Notations)



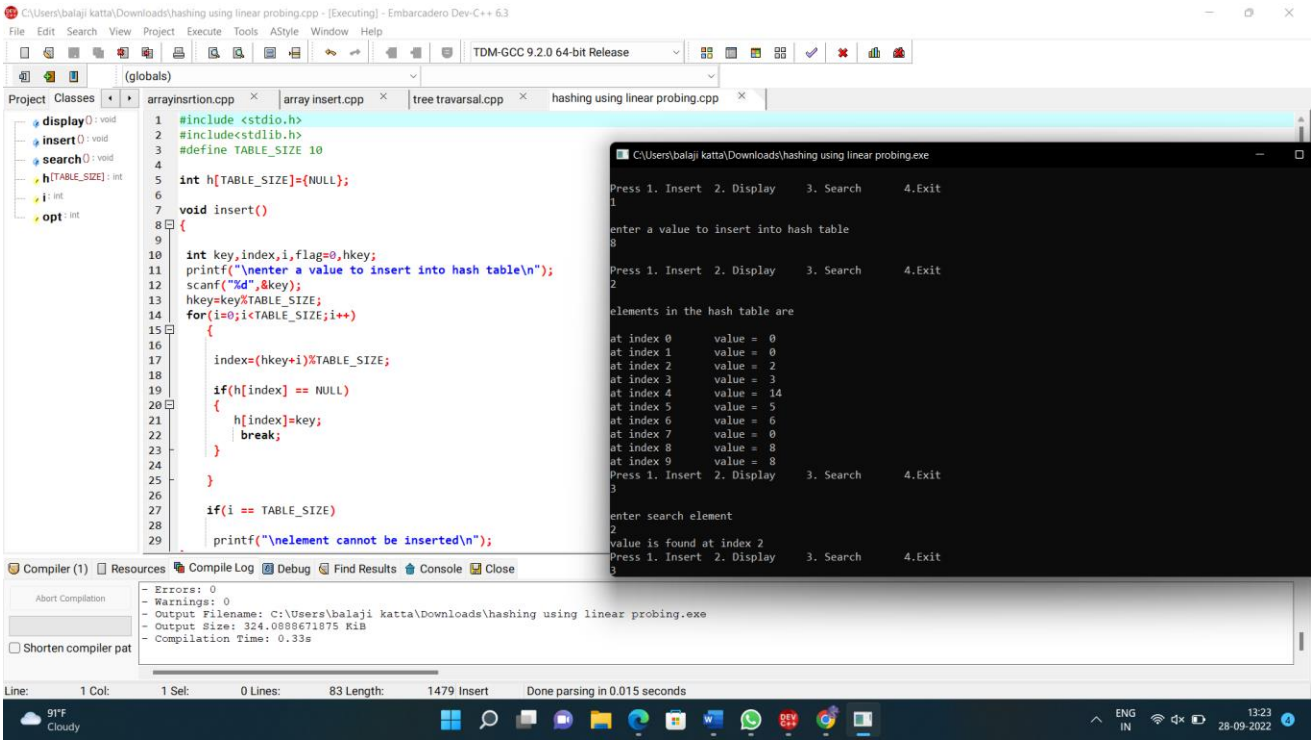
13. Write a C program to implement Queue operations such as ENQUEUE, DEQUEUE and Display



14. Write a C program to implement the Tree Traversals (Inorder, Preorder, Postorder)



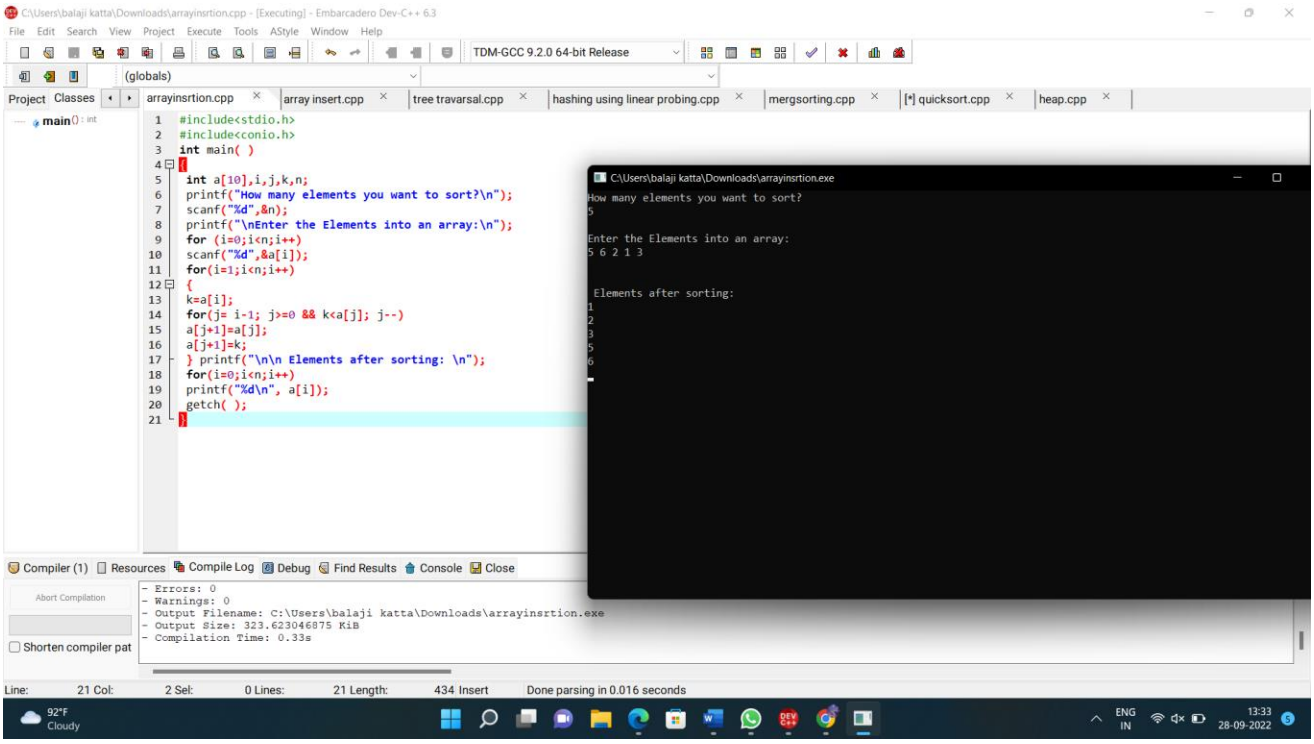
15. Write a C program to implement hashing using Linear Probing method



```
#include <stdio.h>
#include<stdlib.h>
#define TABLE_SIZE 10

int h[TABLE_SIZE]={NULL};

void insert()
{
  int key,index,i,flag=0,hkey;
  printf("\nenter a value to insert into hash table\n");
  scanf("%d",&key);
  hkey=key%TABLE_SIZE;
  for(i=0;i<TABLE_SIZE;i++)
    {

     index=(hkey+i)%TABLE_SIZE;

     if(h[index] == NULL)
     {
        h[index]=key;
         break;
     }

    }

    if(i == TABLE_SIZE)

     printf("\nelement cannot be inserted\n");
```

```
C:\Users\balaji katta\Downloads\hashing using linear probing.exe

Press 1. Insert  2. Display     3. Search       4.Exit
1

enter a value to insert into hash table
8

Press 1. Insert  2. Display     3. Search       4.Exit
2

elements in the hash table are

at index 0       value =  0
at index 1       value =  0
at index 2       value =  2
at index 3       value =  3
at index 4       value =  14
at index 5       value =  5
at index 6       value =  6
at index 7       value =  0
at index 8       value =  8
at index 9       value =  8
Press 1. Insert  2. Display     3. Search       4.Exit
3

enter search element
2
value is found at index 2
Press 1. Insert  2. Display     3. Search       4.Exit
3
```

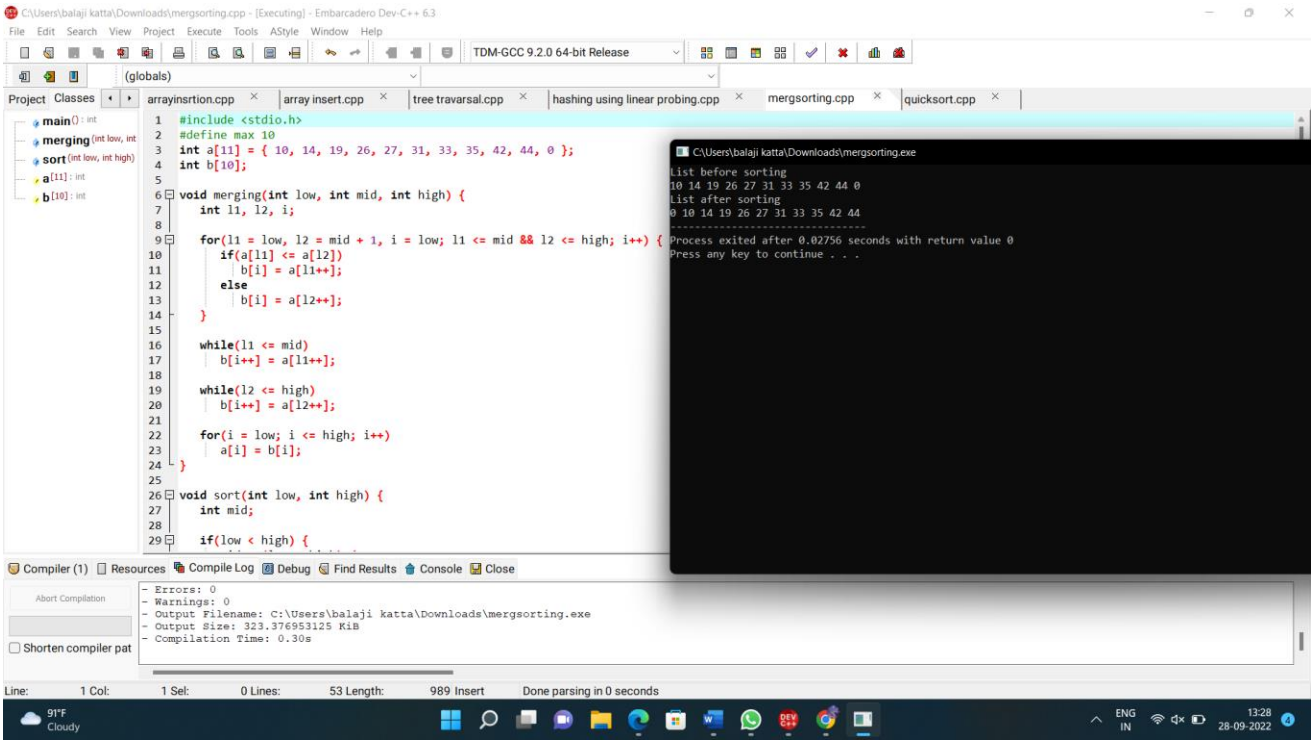
16. Write a C program to arrange a series of numbers using Insertion Sort



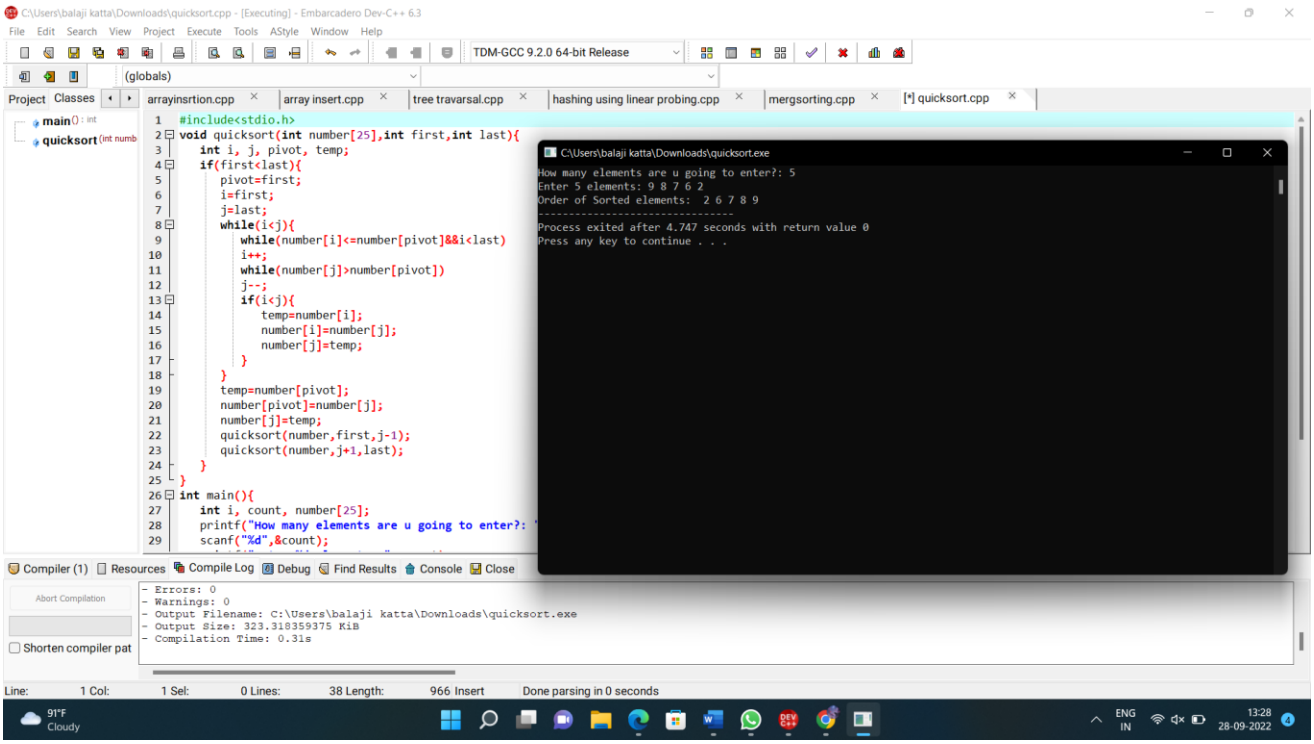
```
#include<stdio.h>
#include<conio.h>
int main( )
{
  int a[10],i,j,k,n;
  printf("How many elements you want to sort?\n");
  scanf("%d",&n);
  printf("\nEnter the Elements into an array:\n");
  for (i=0;i<n;i++)
  scanf("%d",&a[i]);
  for(i=1;i<n;i++)
  {
  k=a[i];
  for(j= i-1; j>=0 && k<a[j]; j--)
  a[j+1]=a[j];
  a[j+1]=k;
  } printf("\n\n Elements after sorting: \n");
  for(i=0;i<n;i++)
  printf("%d\n", a[i]);
  getch( );
}
```

```
C:\Users\balaji katta\Downloads\arrayinsrtion.exe
How many elements you want to sort?
5

Enter the Elements into an array:
5 6 2 1 3


 Elements after sorting:
1
2
3
5
6
```

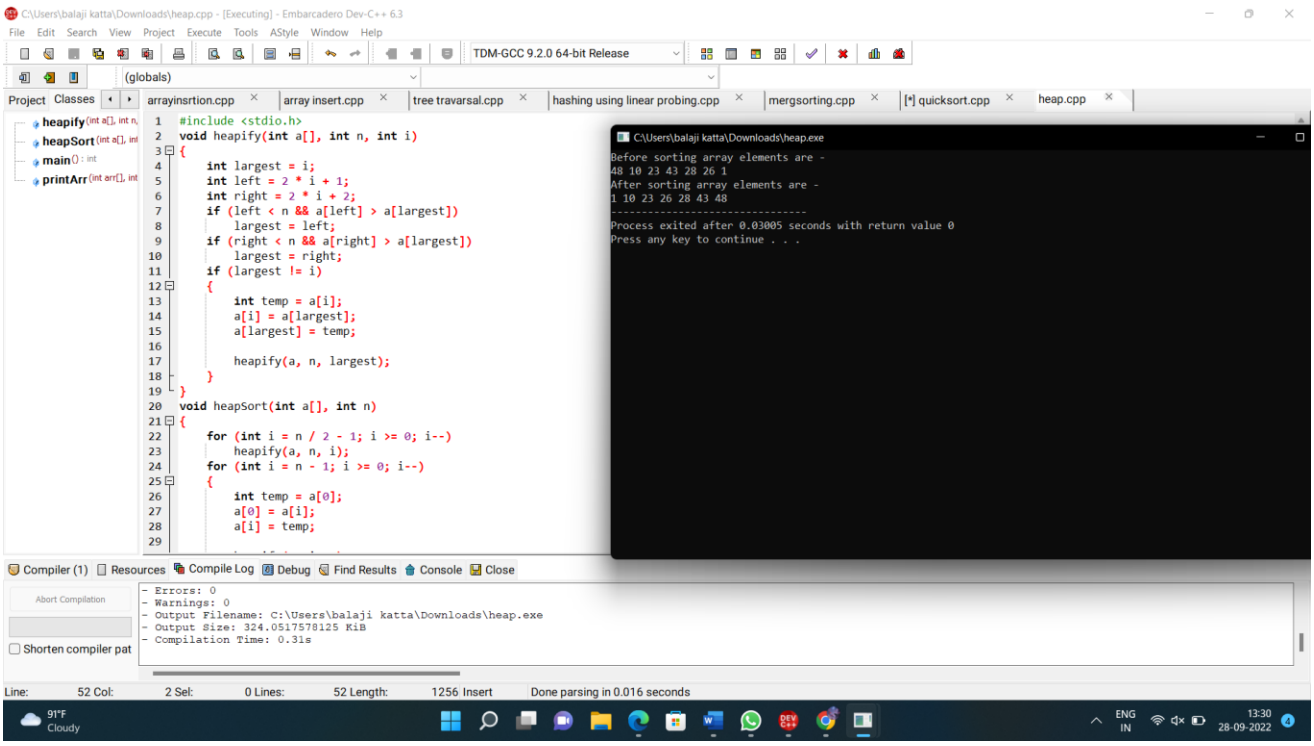
17. Write a C program to arrange a series of numbers using Merge Sort



18. Write a C program to arrange a series of numbers using Quick Sort



19. Write a C program to implement Heap sort.

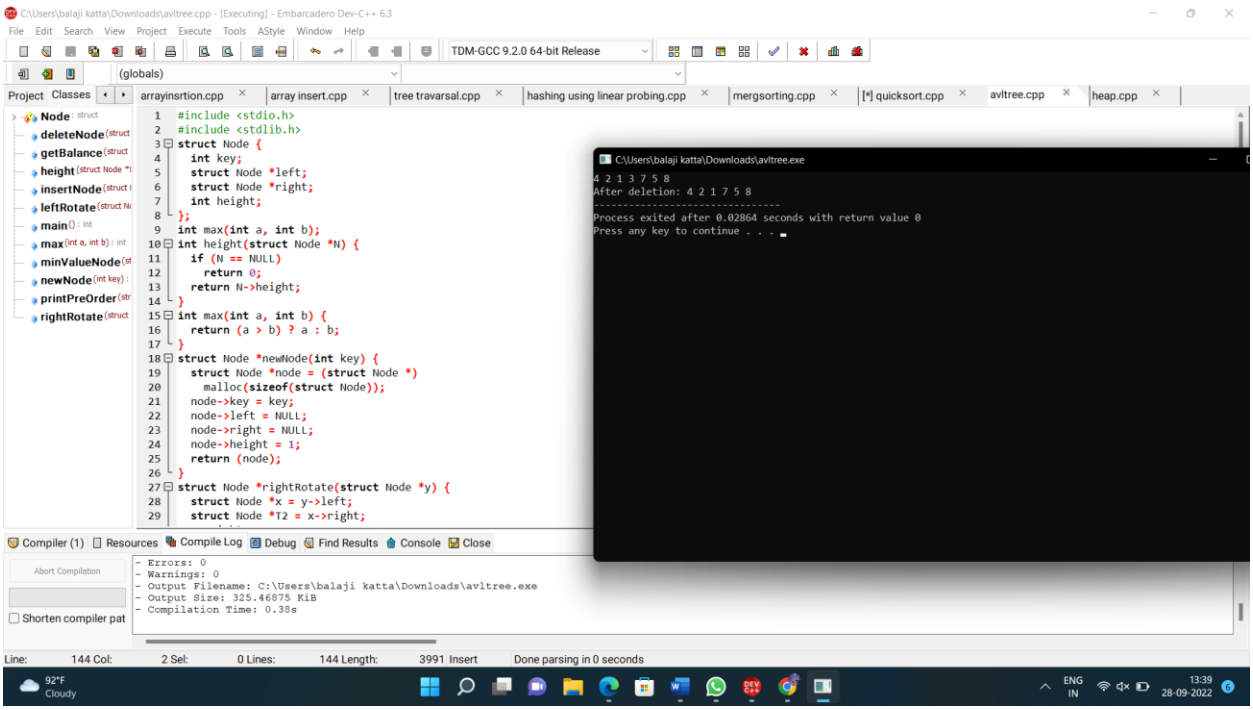


20. Write a program to perform the following operations:

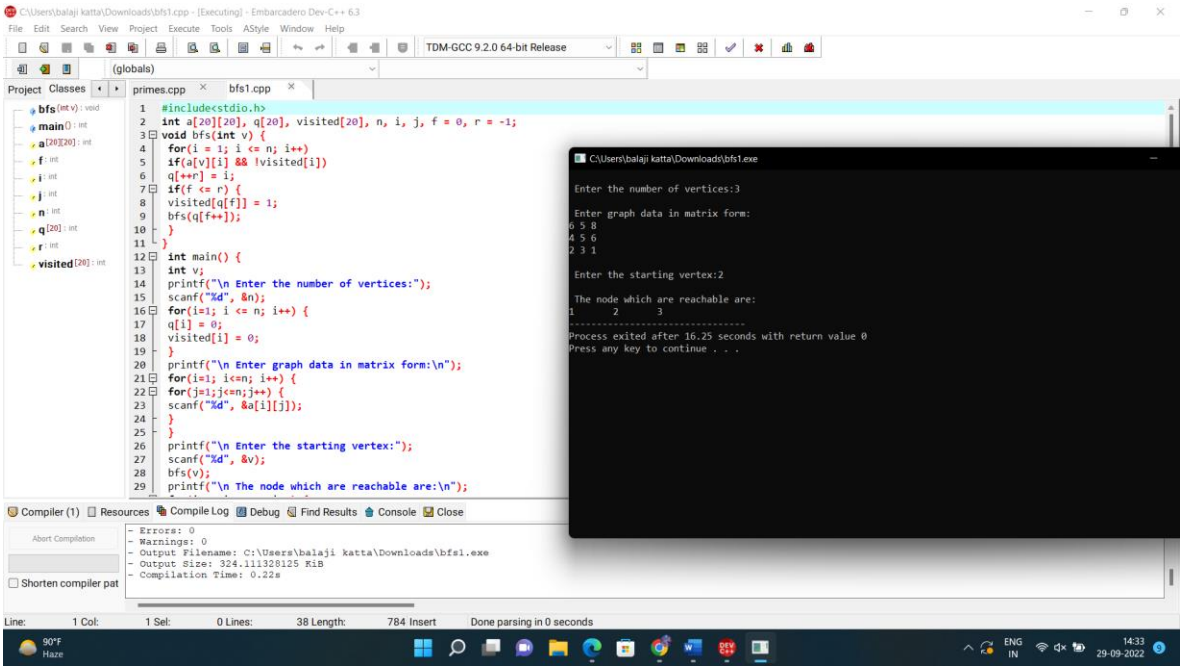
    a) Insert an element into a AVL tree.

    b) Delete an element from a AVL tree.

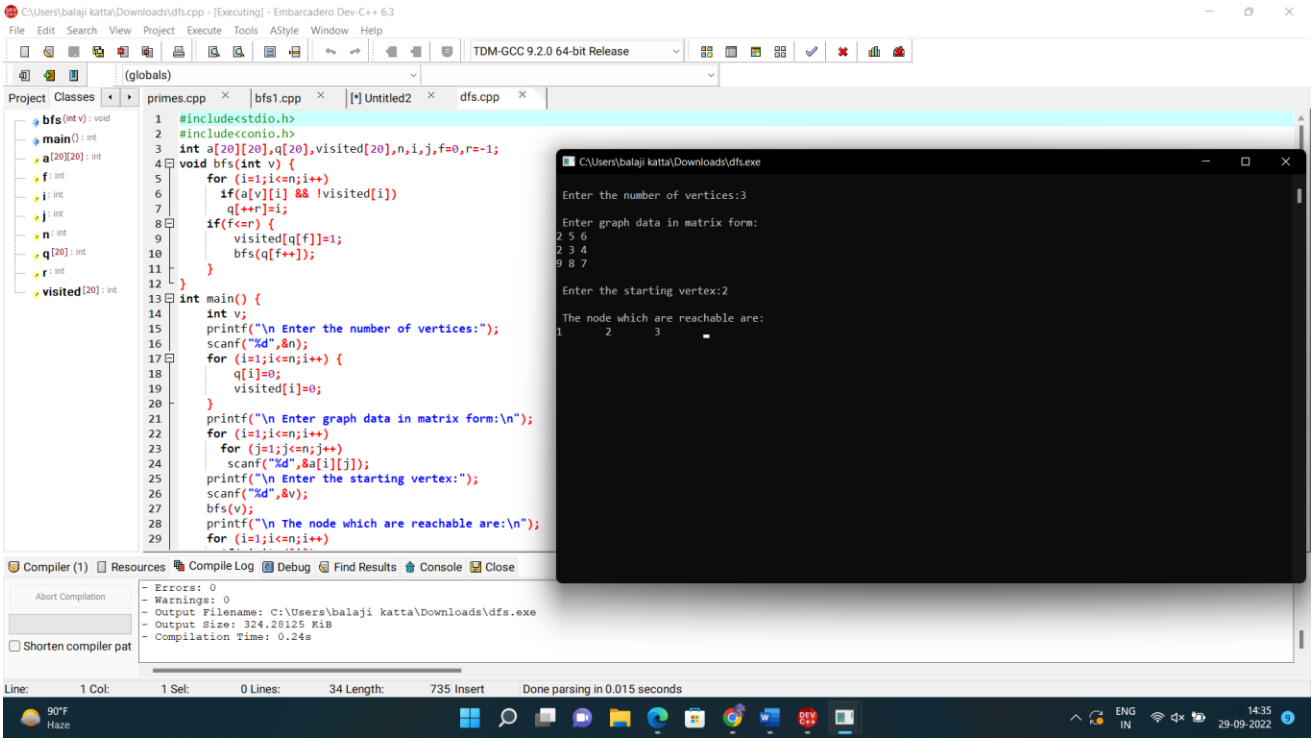
    c) Search for a key element in a AVL tree.



21. Write a C program to Graph traversal using Breadth First Search



22. Write a C program to Graph traversal using Depth First Search

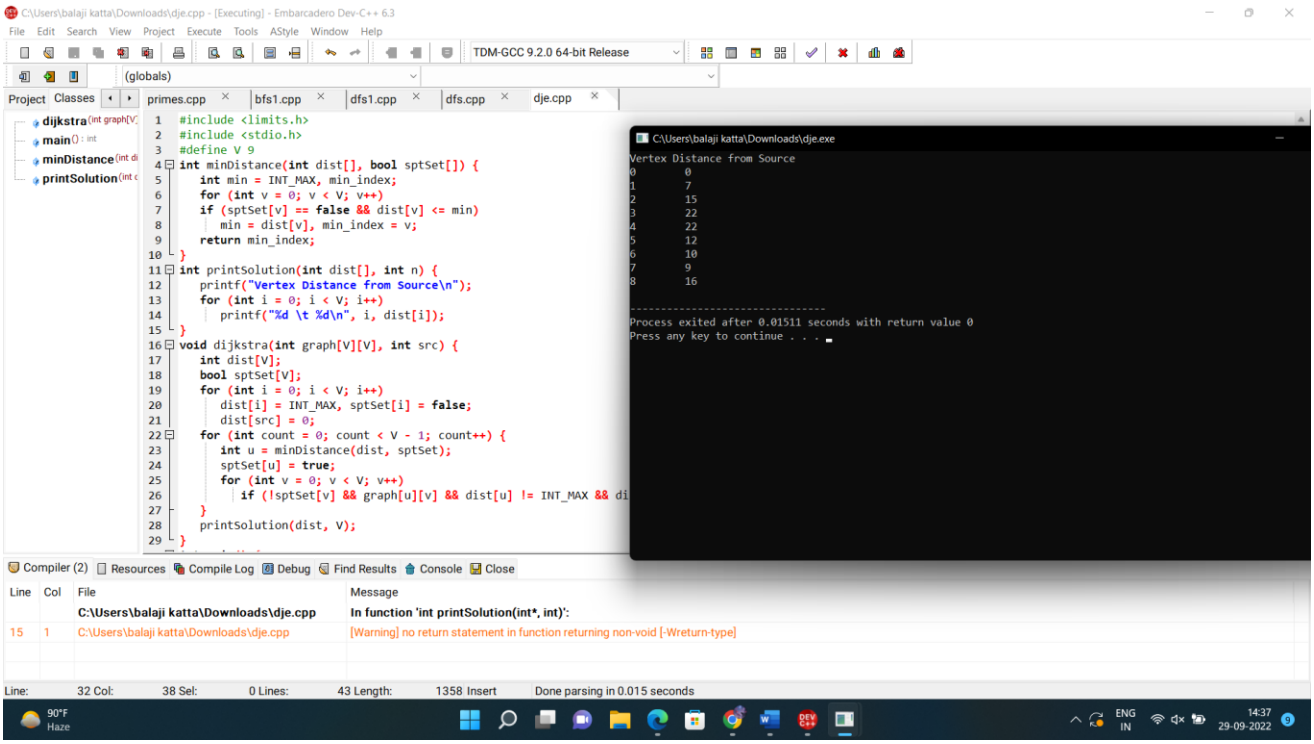


```
#include<stdio.h>
#include<conio.h>
int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
void bfs(int v) {
    for (i=1;i<=n;i++)
      if(a[v][i] && !visited[i])
        q[++r]=i;
    if(f<=r) {
        visited[q[f]]=1;
        bfs(q[f++]);
    }
}
int main() {
    int v;
    printf("\n Enter the number of vertices:");
    scanf("%d",&n);
    for (i=1;i<=n;i++) {
        q[i]=0;
        visited[i]=0;
    }
    printf("\n Enter graph data in matrix form:\n");
    for (i=1;i<=n;i++)
      for (j=1;j<=n;j++)
       scanf("%d",&a[i][j]);
    printf("\n Enter the starting vertex:");
    scanf("%d",&v);
    bfs(v);
    printf("\n The node which are reachable are:\n");
    for (i=1;i<=n;i++)
```

```
C:\Users\balaji katta\Downloads\dfs.exe

Enter the number of vertices:3

Enter graph data in matrix form:
2 5 6
2 3 4
9 8 7

Enter the starting vertex:2

The node which are reachable are:
1       2       3
```

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\balaji katta\Downloads\dfs.exe
- Output Size: 324.28125 KiB
- Compilation Time: 0.24s
```

23. Implementation of Shortest Path Algorithms using Dijkstra's Algorithm

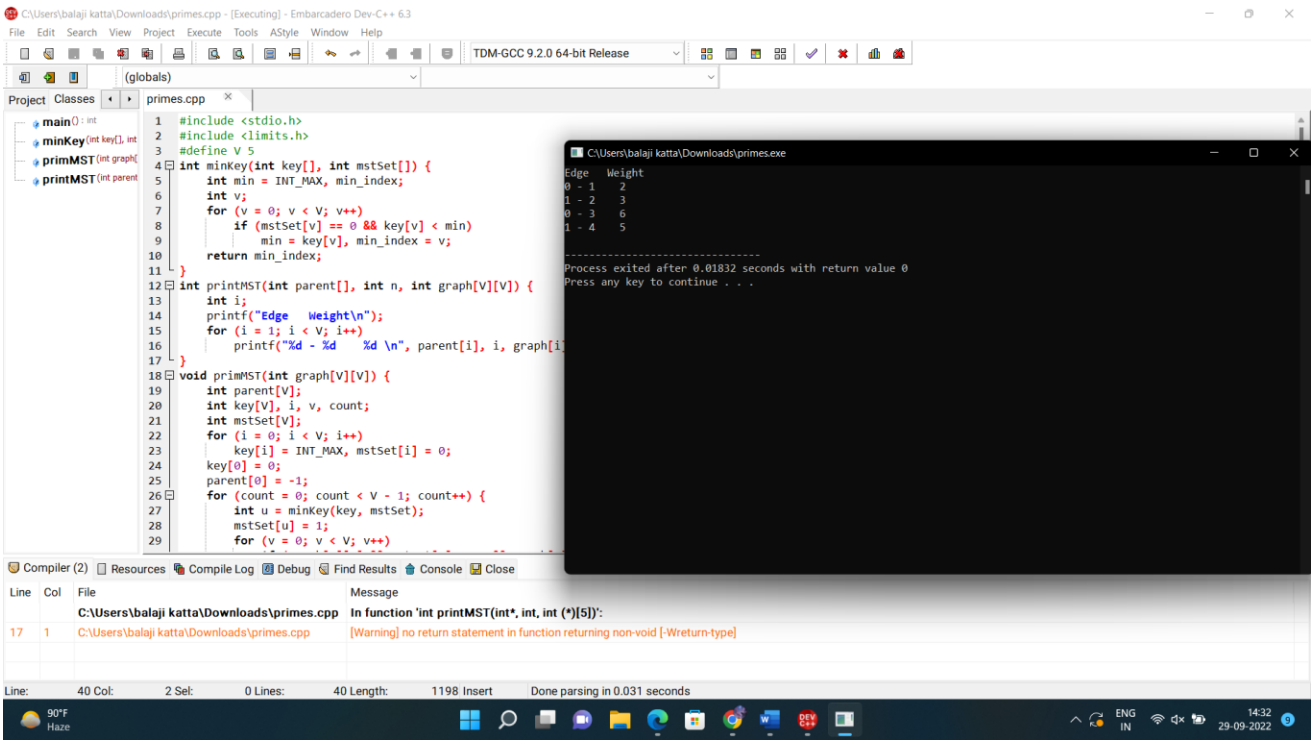


```
#include <limits.h>
#include <stdio.h>
#define V 9
int minDistance(int dist[], bool sptSet[]) {
    int min = INT_MAX, min_index;
    for (int v = 0; v < V; v++)
    if (sptSet[v] == false && dist[v] <= min)
        min = dist[v], min_index = v;
    return min_index;
}
int printSolution(int dist[], int n) {
    printf("Vertex Distance from Source\n");
    for (int i = 0; i < V; i++)
        printf("%d \t %d\n", i, dist[i]);
}
void dijkstra(int graph[V][V], int src) {
    int dist[V];
    bool sptSet[V];
    for (int i = 0; i < V; i++)
        dist[i] = INT_MAX, sptSet[i] = false;
        dist[src] = 0;
    for (int count = 0; count < V - 1; count++) {
        int u = minDistance(dist, sptSet);
        sptSet[u] = true;
        for (int v = 0; v < V; v++)
            if (!sptSet[v] && graph[u][v] && dist[u] != INT_MAX && di
    }
    printSolution(dist, V);
}
```

```
C:\Users\balaji katta\Downloads\dje.exe
Vertex Distance from Source
0       0
1       7
2       15
3       22
4       22
5       12
6       10
7       9
8       16

--------------------------------
Process exited after 0.01511 seconds with return value 0
Press any key to continue . . .
```

Line	Col	File	Message
		C:\Users\balaji katta\Downloads\dje.cpp	In function 'int printSolution(int*, int)':
15	1	C:\Users\balaji katta\Downloads\dje.cpp	[Warning] no return statement in function returning non-void [-Wreturn-type]

24. Implementation of Minimum Spanning Tree using Prim's Algorithm



25. Implementation of Minimum Spanning Tree using Kruskal Algorithm