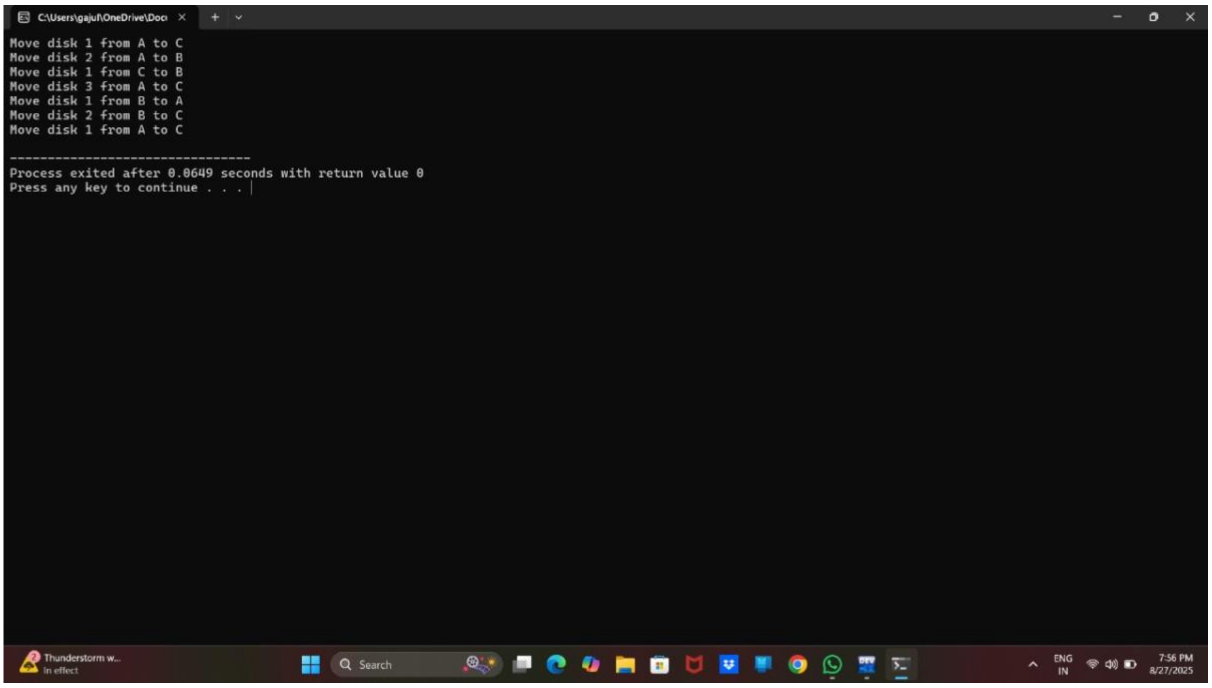
1)(a)Recursive

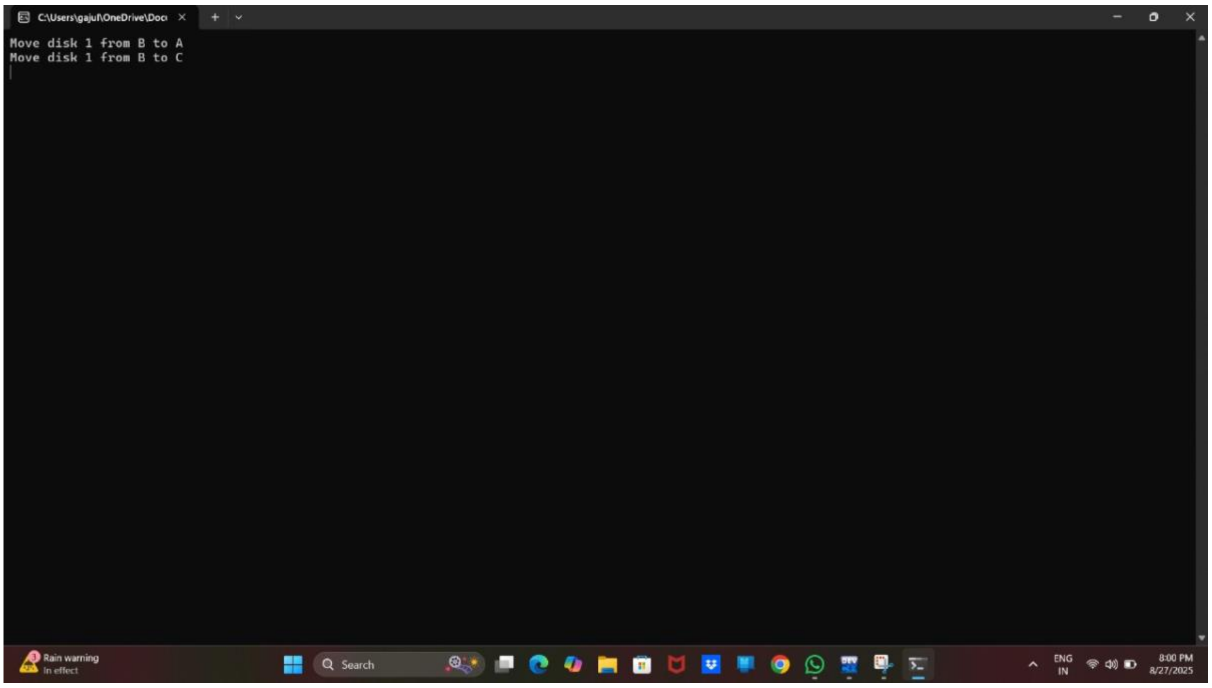
```
#include <stdio.h>


Void hanoi(int n, char from, char to, char aux) {
   If (n == 1) {
      Printf("Move disk 1 from %c to %c\n", from, to);
      Return;
   }
   Hanoi(n – 1, from, aux, to);
   Printf("Move disk %d from %c to %c\n", n, from, to);
   Hanoi(n – 1, aux, to, from);
}


Int main()
 {
   Int n = 3;
   Hanoi(n, 'A', 'C', 'B');
   Return 0;
}
```



```
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C

--------------------------------
Process exited after 0.0649 seconds with return value 0
Press any key to continue . . .
```

(b) Iterative

```
#include <stdio.h>


Typedef struct {
   Int n;
   Char from, to, aux;
   Int stage;
}Frame;


Void hanoi_iterative(int n, char from, char to, char aux) {
   Frame stack[100];
   Int top = -1;


   Stack[++top] = (Frame){n, from, to, aux, 0};


   While (top >= 0) {
        Frame *f = &stack[top--];


        If (f->n == 1) {
            Printf(“Move disk 1 from %c to %c\n”, f->from, f->to);
            Continue;
        }


        If (f->stage == 0)
        {


            Stack[++top] = (Frame){f->n – 1, f->aux, f->to, f->from, 0};
            Stack[++top] = (Frame){1, f->from, f->to, f->aux, 0};
            Stack[++top] = (Frame){f->n – 1, f->from, f->aux, f->to, 0};
        }
    }
}


Int main()
 {
    Int n = 3;
    Hanoi_iterative(n, ‘A’, ‘C’, ‘B’);
    Return 0;
}
```

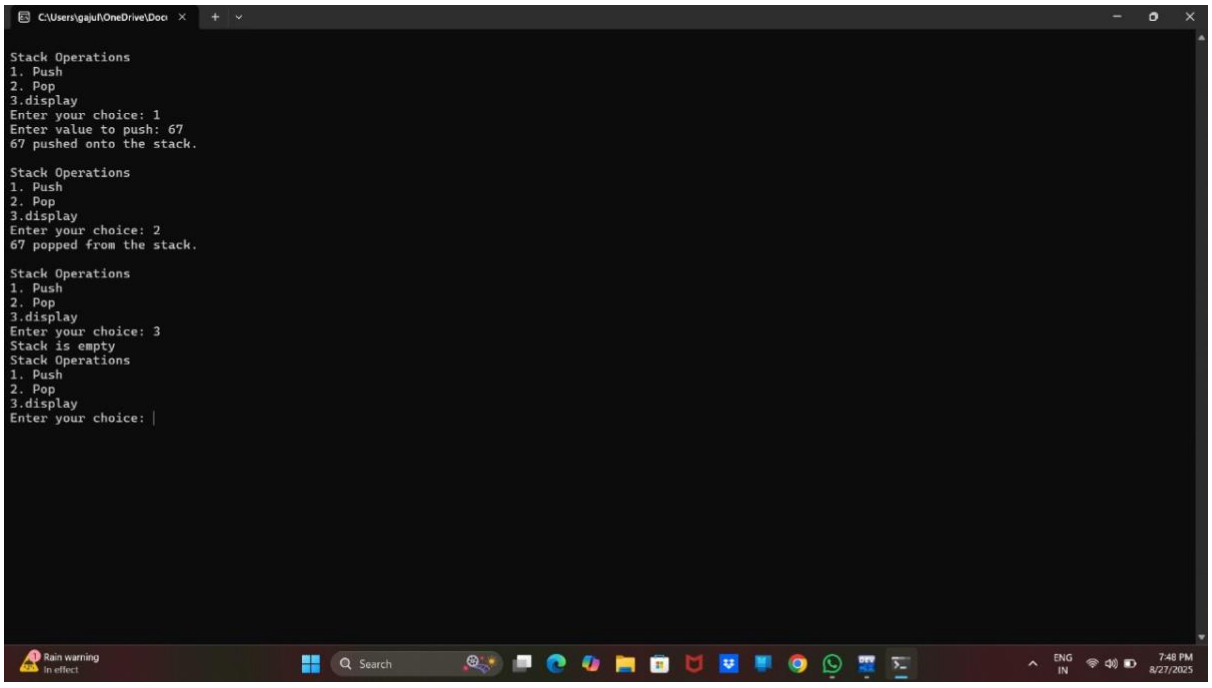


```
Move disk 1 from B to A
Move disk 1 from B to C
```

2) Stack implementation

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 100


Int stack[MAX];

Int top = -1;


Void push(int value)

 {

   If (top == MAX – 1)

   {

      Printf(“Stack Overflow “);

   }

   Else

    {
        Stack[++top] = value;
        Printf(“%d pushed onto the stack.\n”, value);
    }
}

Void pop()
 {
    If (top == -1)
     {
        Printf(“Stack Underflow\n”);
    }
     Else
      {
        Printf(“%d popped from the stack.\n”, stack[top--]);
    }
}

Void display()
 {
    If (top == -1)
     {
        Printf(“Stack is empty “);
    }
    Else
     {
        Printf(“Stack elements are: “);
        For (int i = 0; i <= top; i++)
         {
          Printf("%d ", stack[i]);
        }
        Printf("\n");
    }
}

Int main()
 {
    Int choice, data;

    While (1)
     {
        Printf("\nStack Operations\n");
        Printf("1. Push\n");
        Printf("2. Pop\n");
        Printf("3.display\n");
        Printf("Enter your choice: ");
        Scanf("%d", &choice);

        Switch (choice) {
          Case 1:
            Printf("Enter value to push: ");
            Scanf("%d", &data);
            Push(data);
            Break;
          Case 2:
            Pop();
            Break;
```

```
      Case 3:
        Display();
        Break;

      Default:
        Printf("Invalid choice! Please try again.\n");
    }
  }

  Return 0;
}
```



```
Stack Operations
1. Push
2. Pop
3.display
Enter your choice: 1
Enter value to push: 67
67 pushed onto the stack.

Stack Operations
1. Push
2. Pop
3.display
Enter your choice: 2
67 popped from the stack.

Stack Operations
1. Push
2. Pop
3.display
Enter your choice: 3
Stack is empty
Stack Operations
1. Push
2. Pop
3.display
Enter your choice:
```