

1.STACK IMPLEMENTATION

```
#include<stdio.h>
# define Max 100
int stack[Max];
int top=-1;
void push(int val)
{
    if(top>=Max-1)
        printf("stack overflow");
    else{
        stack[++top]=val;
    }
}

int pop()
{
    if (top<0)
        printf("stack underflow");
    else{
        return stack[top--];
    }
}

int peek()
{
    if(top<0)
        printf("stack is empty");
    else
        return stack[top];
}

void display()
{
    if(top<0)
        printf("stack is empty");
    else{
        printf("stack elements are:");
        for(int i=top;i>=0;i--)
            printf("%d",stack[i]);

    }
}

int main()
{
    push(7);
```

```

    push(6);
    push(32);
    push(50);
    printf("%d is the top element\n",peek());
    printf("%d is popped element\n",pop());
    display( );
    return 0;
}

```

```

C:\Users\gajuhOneDrive\Docu x + v
50 is the top element
50 is popped element
stack elements are:3267
-----
Process exited after 0.09118 seconds with return value 0
Press any key to continue . . .

```

2. Towers of hanoi (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);
}

```

```

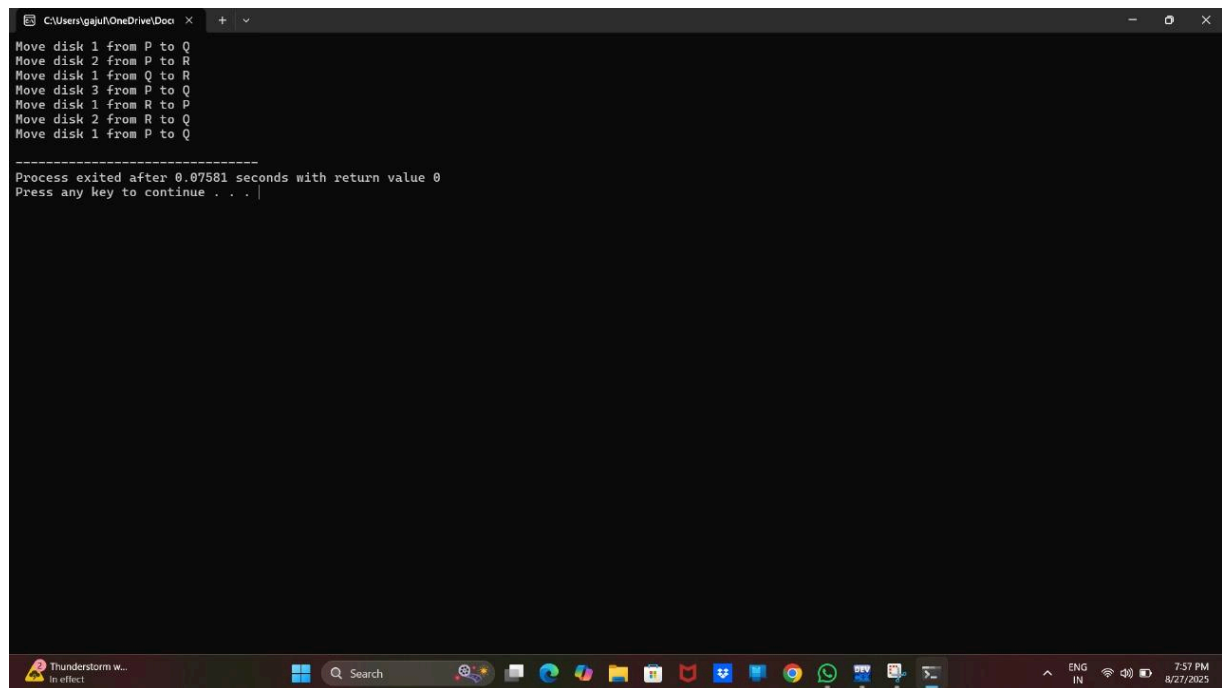
int main()
{
    int n = 3;

```

```

    hanoi(n, 'A', 'C', 'B');
    return 0;
}

```



```

C:\Users\gajuh\OneDrive\Doc
Move disk 1 from P to Q
Move disk 2 from P to R
Move disk 1 from Q to R
Move disk 3 from P to Q
Move disk 1 from R to P
Move disk 2 from R to Q
Move disk 1 from P to Q

-----
Process exited after 0.07581 seconds with return value 0
Press any key to continue . . .

```

3. Towers of hanoi (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;
}

```

