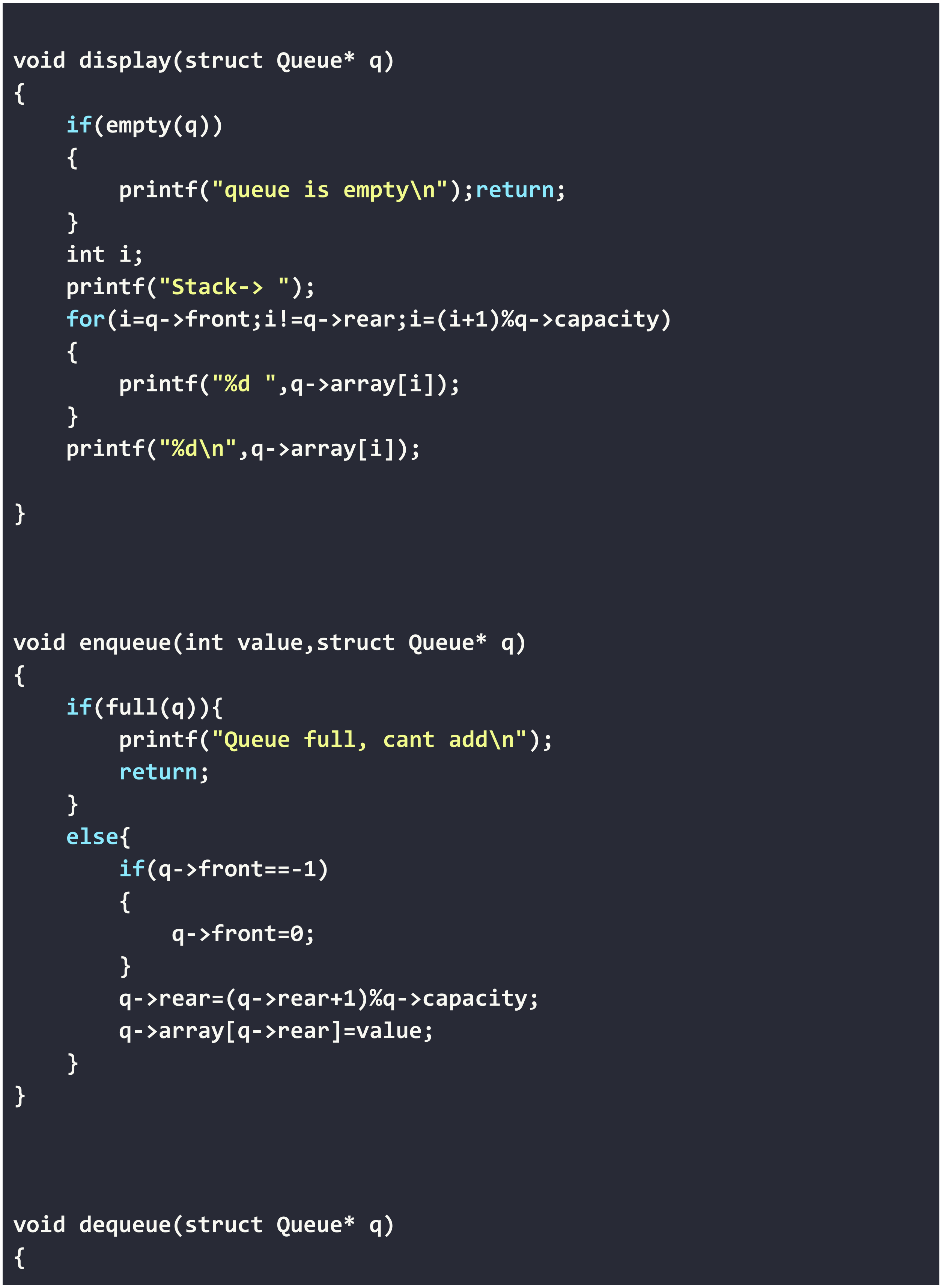
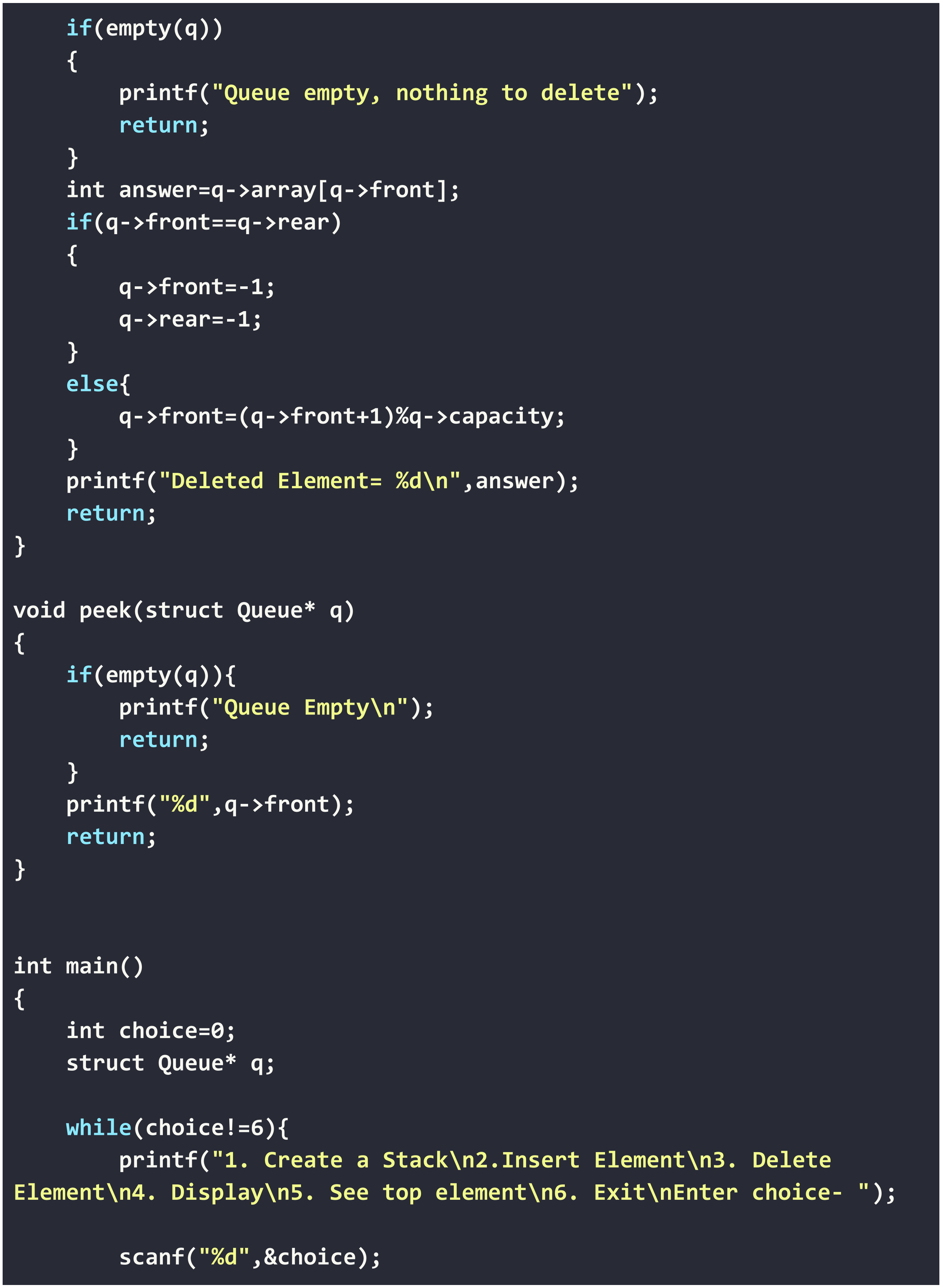
Name- Arya Nair
Roll Number- 16010421063
Batch A2

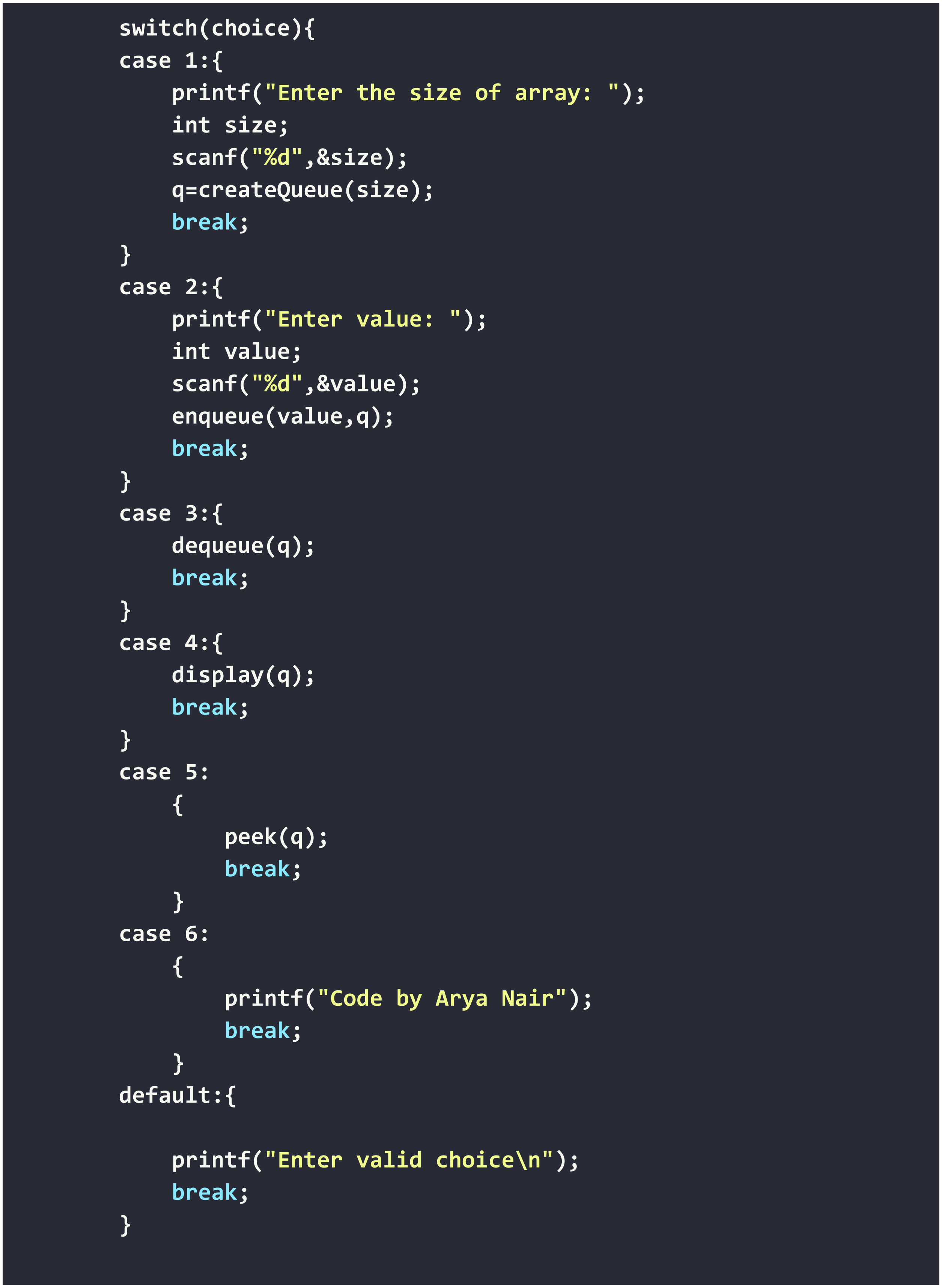
```
#include<stdio.h>

struct Queue
{
    int front;
    int rear;
    unsigned capacity;
    int* array;
};

int full(struct Queue* q)
{
    if((q->front==q->rear+1)|| (q->front ==0 &&
q->rear==q->capacity-1)){
        return 1;
    }
    return 0;
}
int empty(struct Queue* q)
{
    if(q->front==-1)
    {
        return 1;
    }
    return 0;
}

struct Queue* createQueue(unsigned capacity)
{
    struct Queue* queue=(struct Queue*)malloc(sizeof(struct Queue));
    queue->front=-1;
    queue->rear=-1;
    queue->capacity=capacity;
    queue->array=(int*)malloc(capacity*sizeof(int));
    return queue;
};
```



```
void display(struct Queue* q)
{
    if(empty(q))
    {
        printf("queue is empty\n");return;
    }
    int i;
    printf("Stack-> ");
    for(i=q->front;i!=q->rear;i=(i+1)%q->capacity)
    {
        printf("%d ",q->array[i]);
    }
    printf("%d\n",q->array[i]);

}



void enqueue(int value,struct Queue* q)
{
    if(full(q)){
        printf("Queue full, cant add\n");
        return;
    }
    else{
        if(q->front==-1)
        {
            q->front=0;
        }
        q->rear=(q->rear+1)%q->capacity;
        q->array[q->rear]=value;
    }
}



void dequeue(struct Queue* q)
{
```

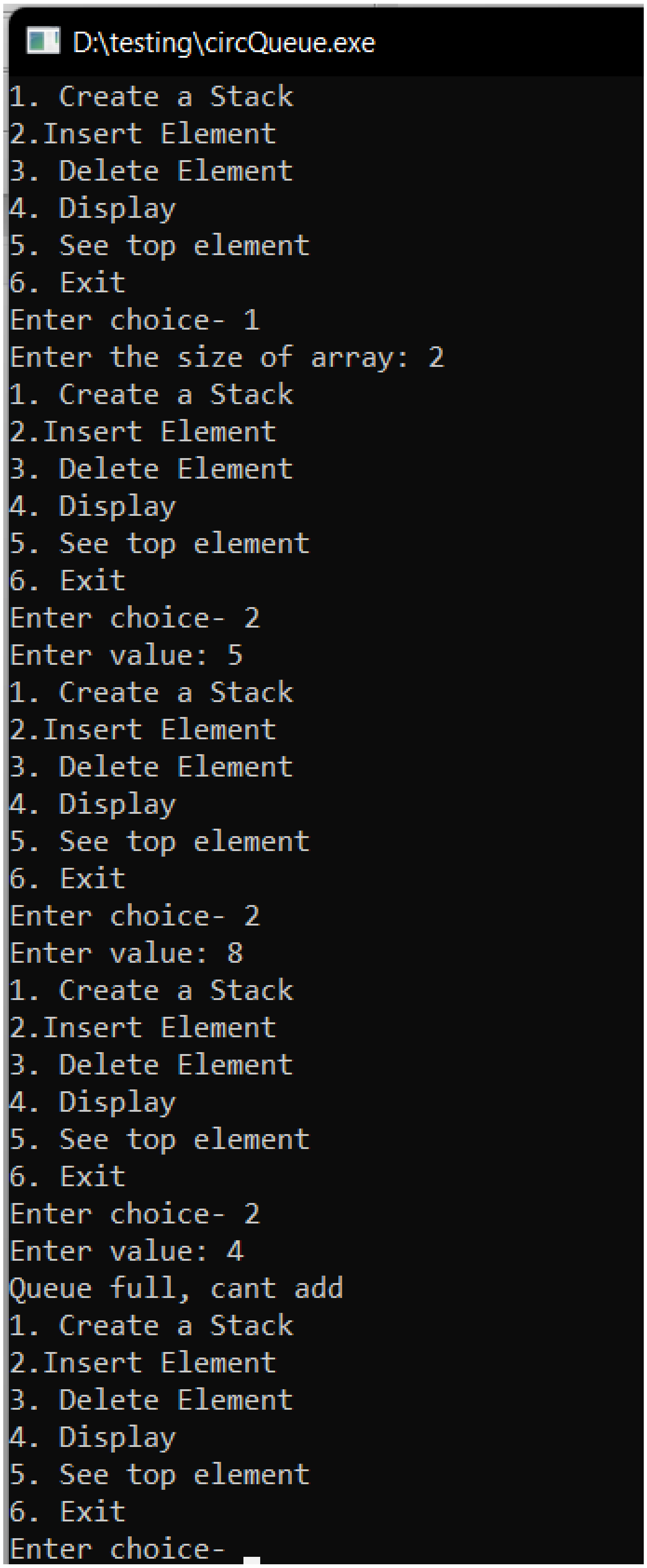


```
    if(empty(q))
    {
        printf("Queue empty, nothing to delete");
        return;
    }
    int answer=q->array[q->front];
    if(q->front==q->rear)
    {
        q->front=-1;
        q->rear=-1;
    }
    else{
        q->front=(q->front+1)%q->capacity;
    }
    printf("Deleted Element= %d\n",answer);
    return;
}

void peek(struct Queue* q)
{
    if(empty(q)){
        printf("Queue Empty\n");
        return;
    }
    printf("%d",q->front);
    return;
}


int main()
{
    int choice=0;
    struct Queue* q;

    while(choice!=6){
        printf("1. Create a Stack\n2.Insert Element\n3. Delete 
Element\n4. Display\n5. See top element\n6. Exit\nEnter choice- ");

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



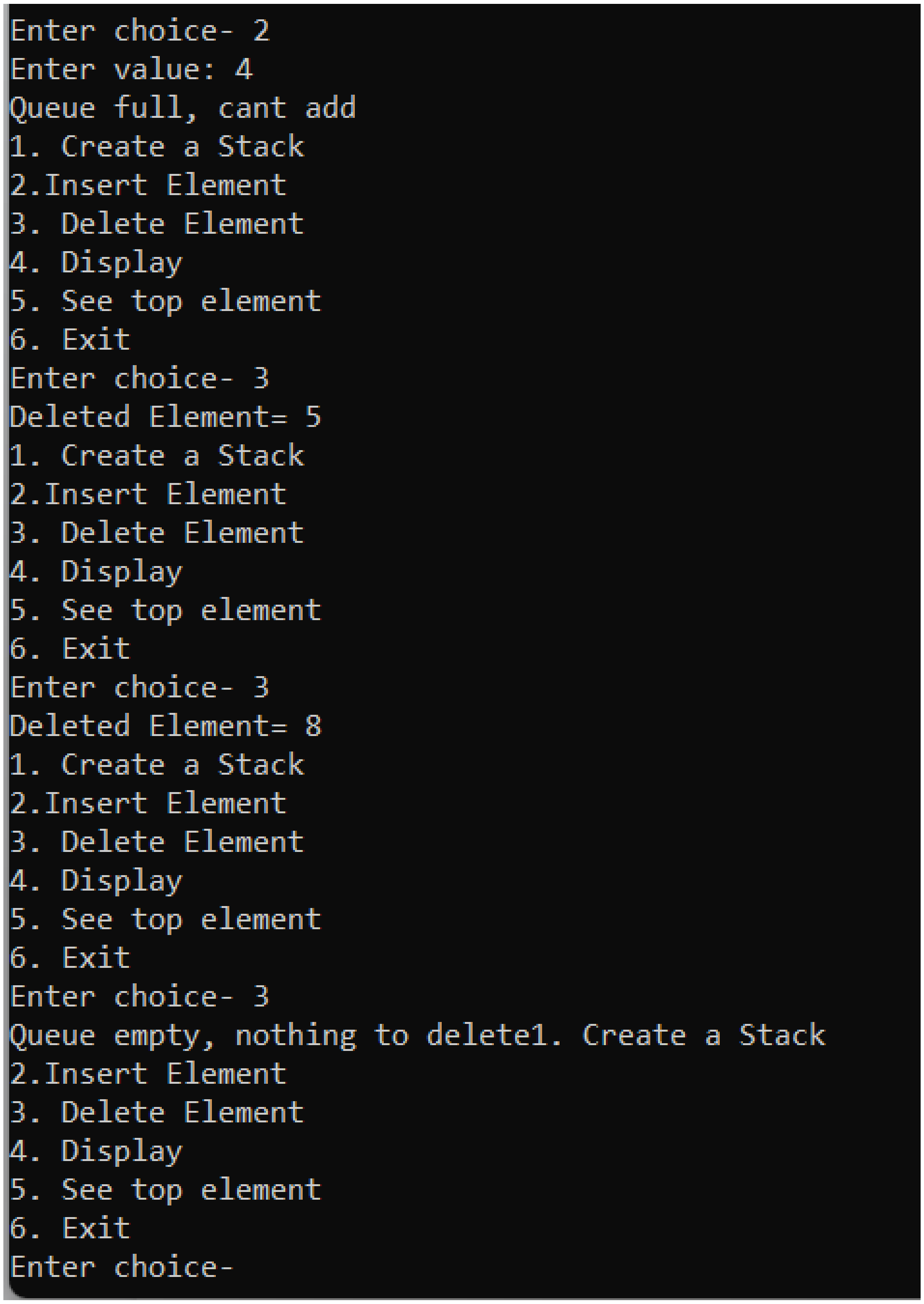
```
switch(choice){
case 1:{
    printf("Enter the size of array: ");
    int size;
    scanf("%d",&size);
    q=createQueue(size);
    break;
}
case 2:{
    printf("Enter value: ");
    int value;
    scanf("%d",&value);
    enqueue(value,q);
    break;
}
case 3:{
    dequeue(q);
    break;
}
case 4:{
    display(q);
    break;
}
case 5:
    {
        peek(q);
        break;
    }
case 6:
    {
        printf("Code by Arya Nair");
        break;
    }
default:{

    printf("Enter valid choice\n");
    break;
}
```

```
        }
    }
}
```

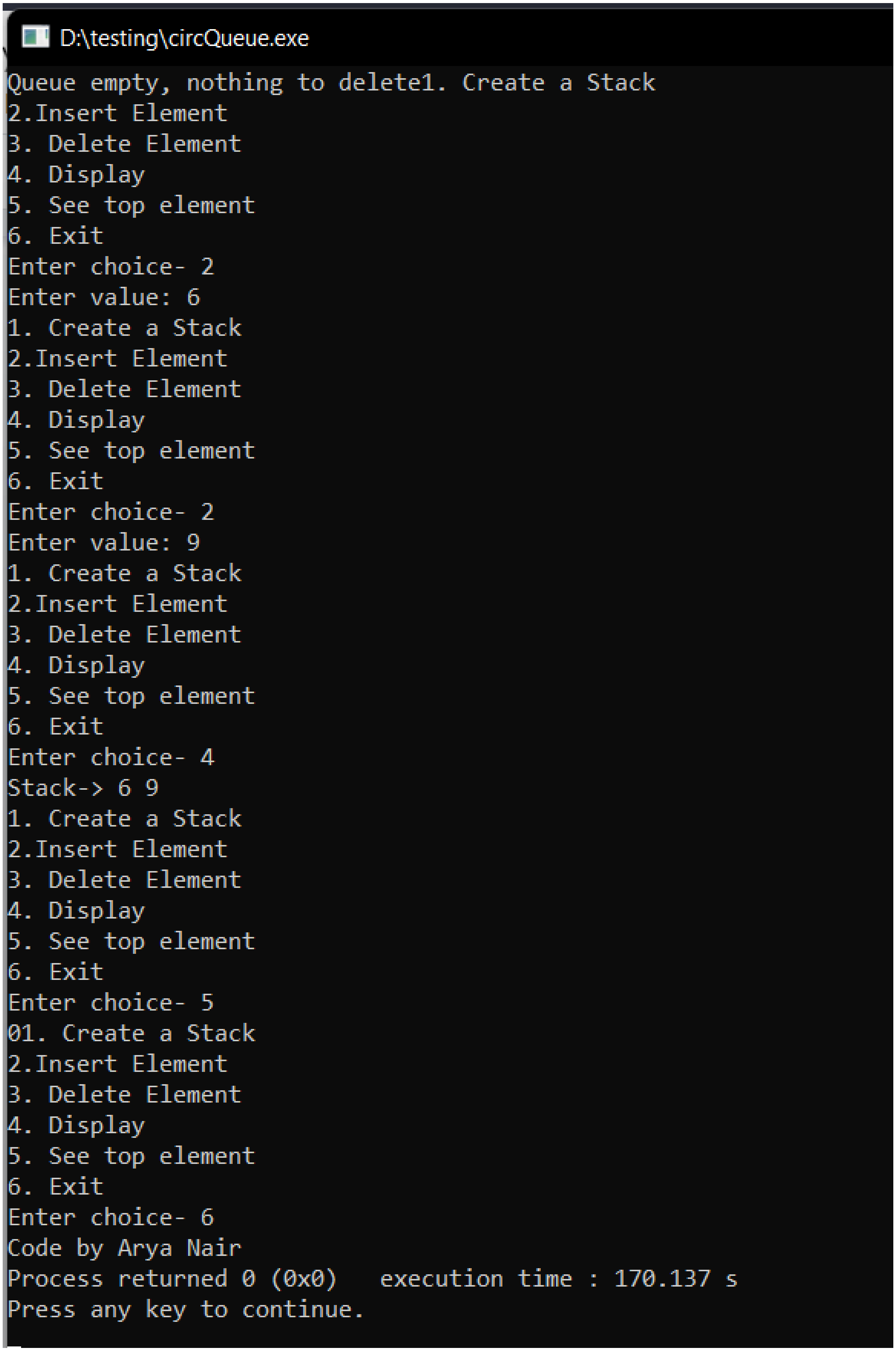


D:\testing\circQueue.exe

```
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 1
Enter the size of array: 2
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 2
Enter value: 5
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 2
Enter value: 8
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 2
Enter value: 4
Queue full, cant add
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- _
```



```
Enter choice- 2
Enter value: 4
Queue full, cant add
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 3
Deleted Element= 5
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 3
Deleted Element= 8
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 3
Queue empty, nothing to delete1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice-
```



```
Queue empty, nothing to delete1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 2
Enter value: 6
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 2
Enter value: 9
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 4
Stack-> 6 9
1. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 5
01. Create a Stack
2.Insert Element
3. Delete Element
4. Display
5. See top element
6. Exit
Enter choice- 6
Code by Arya Nair
Process returned 0 (0x0)   execution time : 170.137 s
Press any key to continue.
```