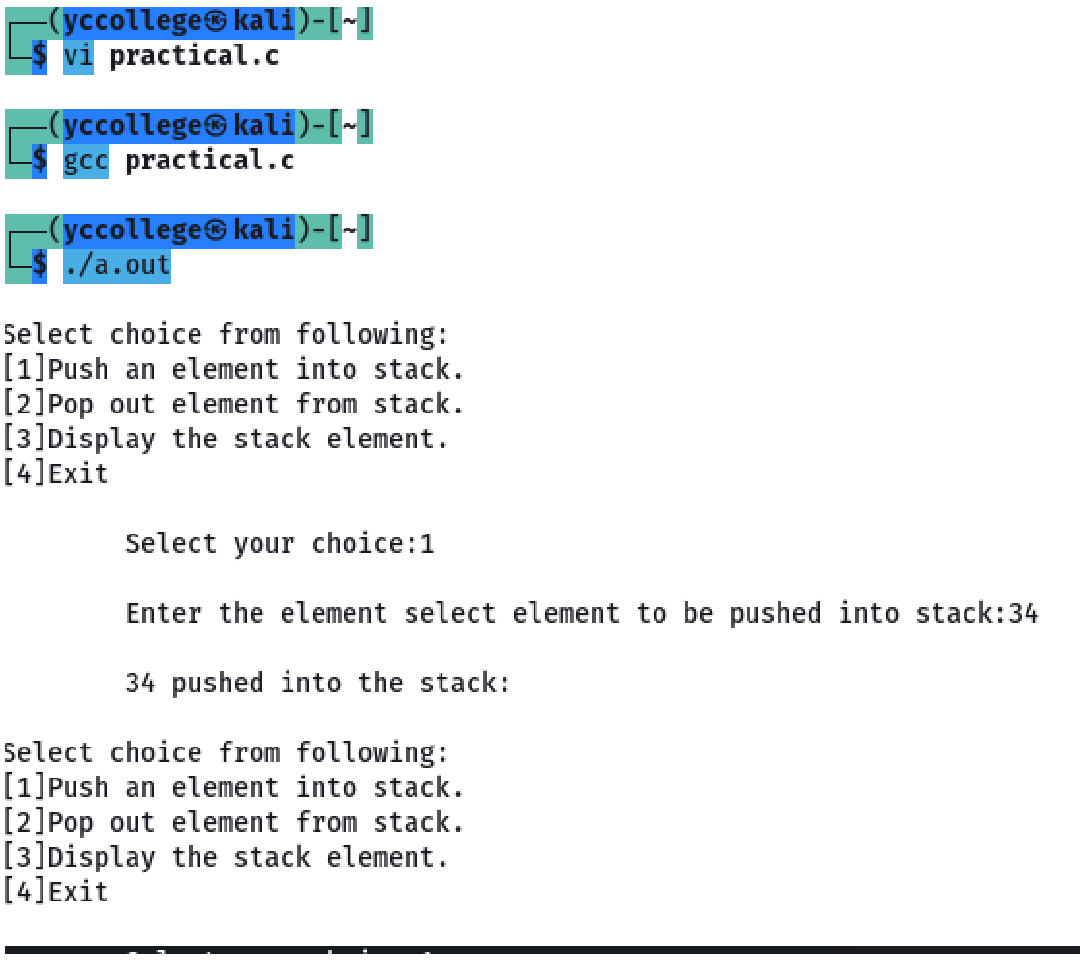
Program:-

```
#include<stdio.h>
#include<stdlib.h>
#define max 10
int stack[max];
int top=-1;
void push(int);
int pop();
void display();
int main()
{
          int choice;
          int num1,num2;
        while(1)
          {
        printf("\nSelect choice from following:");
        printf("\n[1]Push an element into stack.");
        printf("\n[2]Pop out element from stack.");
        printf("\n[3]Display the stack element.");
        printf("\n[4]Exit\n");
        printf("\n\tSelect your choice:");
        scanf("%d",&choice);
        switch(choice)
          {
                    case 1:
                              {
                                        printf("\n\tEnter the element select element to
be pushed into stack:");
                                        scanf("%d",&num1);
                    push(num1);
                    break;
                }
            case 2:
                {
                    num2=pop();
                    printf("\n\t%dElement poped out of the
stack:\n\t",num2);
                    break;
                }
            case 3:
                {
                    display();
                    break;
                }
            case 4:
                {
                    exit(1);
                    break;
                }
            default:
                {
                    printf("\n invalid choice!");
                    break;
                }
        }
        }
        return 0;
}
```

```
void push (int element)
{
        if(top==max-1)
        {
                printf("stack is full");
                exit(1);
        }
        top=top+1;
        stack[top]=element;
        printf("\n\t%d pushed into the stack:\n\t",stack[top]);
}
int pop()
{
        if(top==-1)
        {
                printf("stack is empty");
                exit(1);
        }
        return(stack[top--]);
}
void display()
{
        int i;
        if(top==-1)
        {
                printf("stack is empty");
                exit(1);
        }
        else
```

```
		{
			printf("\n\t The various stack element are:");
			for(i=top;i>=0;i--)
			{
				printf("\t%d",stack[i]);
			}
		}
	}
```

OUTPUT:-



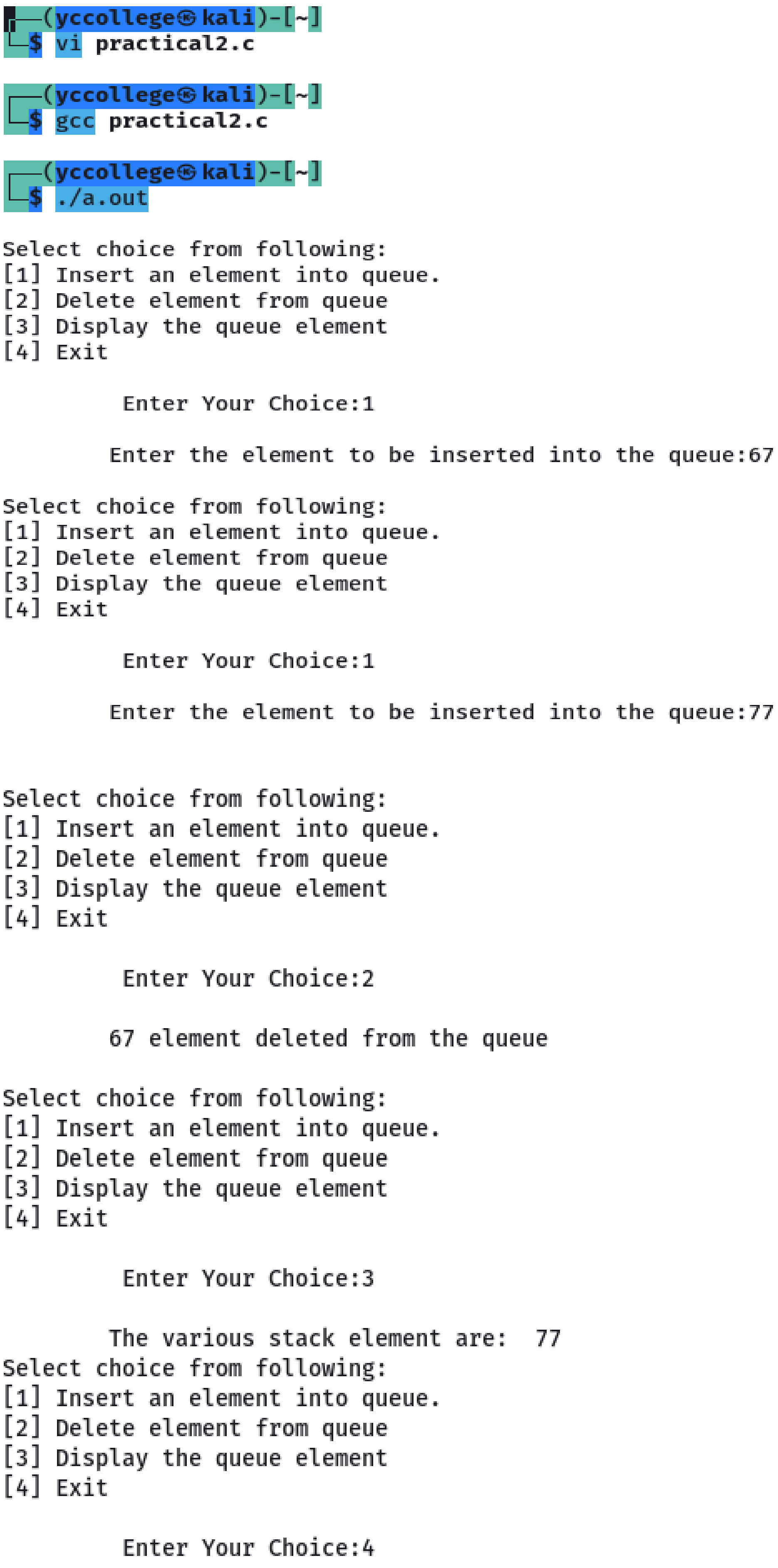
```
┌──(yccollege㉿kali)-[~]
└─$ vi practical.c

┌──(yccollege㉿kali)-[~]
└─$ gcc practical.c

┌──(yccollege㉿kali)-[~]
└─$ ./a.out

Select choice from following:
[1]Push an element into stack.
[2]Pop out element from stack.
[3]Display the stack element.
[4]Exit

        Select your choice:1

        Enter the element select element to be pushed into stack:34

        34 pushed into the stack:

Select choice from following:
[1]Push an element into stack.
[2]Pop out element from stack.
[3]Display the stack element.
[4]Exit
```

```
        Select your choice:1

        Enter the element select element to be pushed into stack:23

        23 pushed into the stack:

Select choice from following:
[1]Push an element into stack.
[2]Pop out element from stack.
[3]Display the stack element.
[4]Exit

        Select your choice:2

        23Element poped out of the stack:

Select choice from following:
[1]Push an element into stack.
[2]Pop out element from stack.
[3]Display the stack element.
[4]Exit

        Select your choice:3

         The various stack element are: 34
Select choice from following:
[1]Push an element into stack.
[2]Pop out element from stack.
[3]Display the stack element.
[4]Exit

        Select your choice:4
```

Program:-

```
#include<stdio.h>
#include<stdlib.h>
#define max 10
int queue[max];
int front=-1;
int rear=-1;
void enqueue(int);
int dequeue();
void display();
int main()

{
    int choice;
    int num1,num2;
    while(1)
   {
      printf("\nSelect choice from following:");
      printf("\n[1] Insert an element into queue.");
      printf("\n[2] Delete element from queue");
      printf("\n[3] Display the queue element");
       printf("\n[4] Exit \n");

      printf("\n\t Enter Your Choice:");
      scanf("%d",&choice);
      switch(choice)
     {
        case 1:
       {
          printf("\n\tEnter the element to be inserted into the queue:");
          scanf("%d",&num1);
          enqueue(num1);
          break;
       }
       case 2:
        {

           num2=dequeue();
         printf("\n\t%d element deleted from the queue\n\t",num2);
         break;
       }
        case 3:
       {
           display();
           break;
      }
      case 4:
      {
          exit(1);
          break;
       }
       default:
       {
          printf("\nInvalid Choice!");
          break;

        }

     }

  }
   return 0;
}
void enqueue(int element)
{
     if(rear==max-1)
    {
       printf("Queue is full.");
       exit(1);
     }
     rear=rear+1;
     queue[rear]=element;
     if(front==-1)
    {
         front=0;
     }
  }
  int dequeue()
  {
     if(front==-1)
    {

           printf("Queue is empty");
           exit(1);
     }

     int data=queue[front];
     queue[front]=0;
     if(front==rear)
    {
        front=rear=-1;
       }
       else
       {
         front++;
       }
       return data;
    }
    void display()
    {
       int i;
       if(front==-1)
       {

           printf("Queue is empty");
           exit(1);
       }
       else
       {

           printf("\n\tThe various stack element are:");
           for(i=front;i<=rear;i++)
         {
             printf("\t%d",queue[i]);

           }

    }


}
```

OUTPUT:



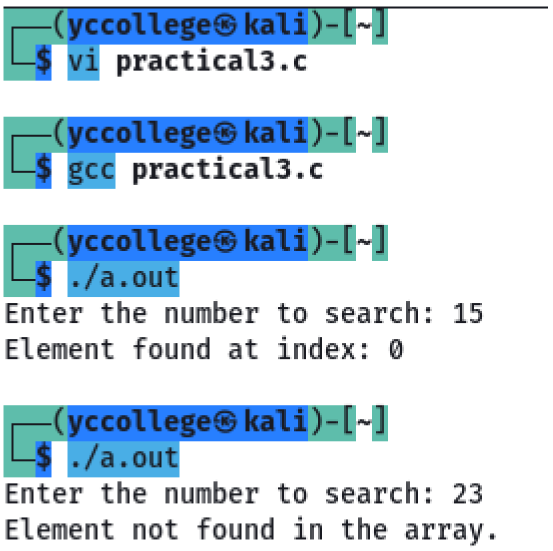
```
┌──(yccollege㉿kali)-[~]
└─$ vi practical2.c

┌──(yccollege㉿kali)-[~]
└─$ gcc practical2.c

┌──(yccollege㉿kali)-[~]
└─$ ./a.out

Select choice from following:
[1] Insert an element into queue.
[2] Delete element from queue
[3] Display the queue element
[4] Exit

         Enter Your Choice:1

        Enter the element to be inserted into the queue:67

Select choice from following:
[1] Insert an element into queue.
[2] Delete element from queue
[3] Display the queue element
[4] Exit

         Enter Your Choice:1

        Enter the element to be inserted into the queue:77


Select choice from following:
[1] Insert an element into queue.
[2] Delete element from queue
[3] Display the queue element
[4] Exit

         Enter Your Choice:2

        67 element deleted from the queue

Select choice from following:
[1] Insert an element into queue.
[2] Delete element from queue
[3] Display the queue element
[4] Exit

         Enter Your Choice:3

        The various stack element are:  77
Select choice from following:
[1] Insert an element into queue.
[2] Delete element from queue
[3] Display the queue element
[4] Exit

         Enter Your Choice:4
```

Program:-

```
#include <stdio.h>
int linearSearch(int arr[], int size, int target) {
    for (int i = 0; i < size; i++)
    {
        if (arr[i] == target)
        {
            return i;
        }
    }
    return -1;
}
int main() {
    int arr[] = {15,20,52,42,60};
    int size = sizeof(arr) / sizeof(arr[0]);
    int target;

    printf("Enter the number to search: ");
    scanf("%d", &target);
    int result = linearSearch(arr, size, target);
    if (result != -1)
    {
        printf("Element found at index: %d\n", result);
    } else
    {
        printf("Element not found in the array.\n");
    }
     return 0;
```

}

OUTPUT:-

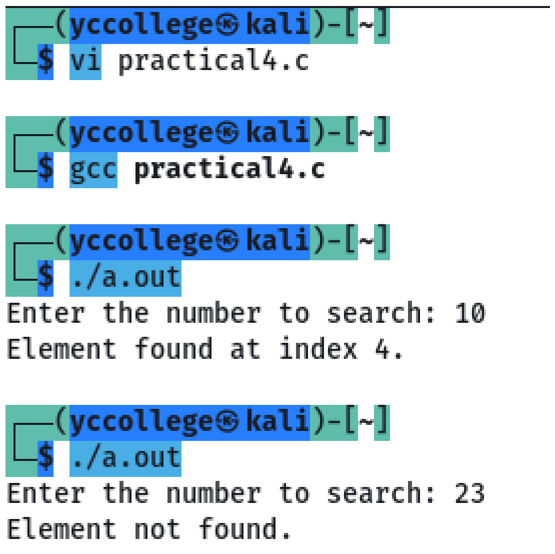


```
┌──(yccollege㉿kali)-[~]
└─$ vi practical3.c

┌──(yccollege㉿kali)-[~]
└─$ gcc practical3.c

┌──(yccollege㉿kali)-[~]
└─$ ./a.out
Enter the number to search: 15
Element found at index: 0

┌──(yccollege㉿kali)-[~]
└─$ ./a.out
Enter the number to search: 23
Element not found in the array.
```

Program:

```
#include <stdio.h>
int binarySearch(int arr[], int size, int target)
{
    int left = 0, right = size - 1;
      while (left <= right) {
        int middle = left + (right - left) / 2;
        if (arr[middle] == target)
            return middle;
         if (arr[middle] < target)
            left = middle + 1;
        else
            right = middle - 1;
    }
    return -1;
}
int main() {
    int arr[] = {2, 4, 6, 8, 10, 14, 18, 30};
    int size = sizeof(arr) / sizeof(arr[0]);
    int target;
    printf("Enter the number to search: ");
    scanf("%d", &target);
   int result = binarySearch(arr, size, target);
     if (result != -1)
        printf("Element found at index %d.\n", result);
    else
        printf("Element not found.\n");
       return 0;
}
```

OUTPUT:-

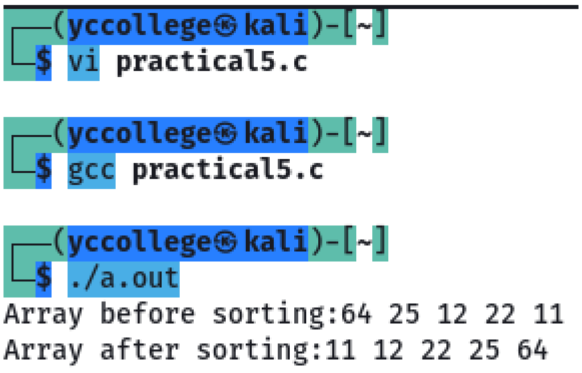


```
┌──(yccollege㉿kali)-[~]
└─$ vi practical4.c

┌──(yccollege㉿kali)-[~]
└─$ gcc practical4.c

┌──(yccollege㉿kali)-[~]
└─$ ./a.out
Enter the number to search: 10
Element found at index 4.

┌──(yccollege㉿kali)-[~]
└─$ ./a.out
Enter the number to search: 23
Element not found.
```

Program:

```
#include <stdio.h>
void selectionSort(int arr[],int n)
{
    int i, j, minIdx, temp;
    for (i = 0; i < n - 1; i++)
    {
        minIdx = i;
        for (j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[minIdx]) a
            {
                minIdx = j;
            }
        }
        temp = arr[minIdx];
        arr[minIdx] = arr[i];
        arr[i] = temp;
    }
}
int main()
{
    int i;
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("Array before sorting:");
    for (i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
    selectionSort(arr, n);
    printf("Array before sorting:");
    for (i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");



    return 0;
}
```

OUTPUT:-

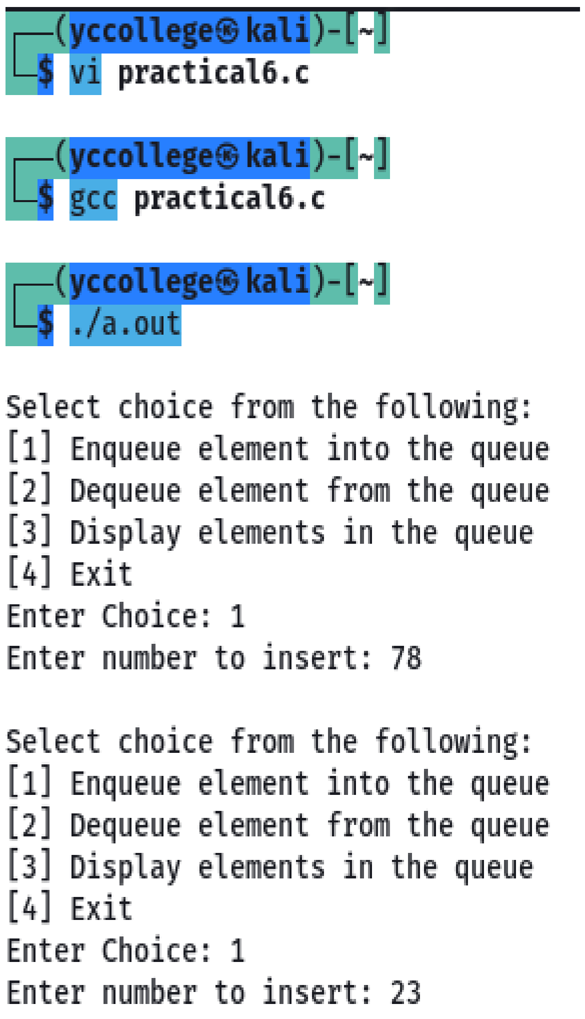


```
┌──(yccollege㉿kali)-[~]
└─$ vi practical5.c

┌──(yccollege㉿kali)-[~]
└─$ gcc practical5.c

┌──(yccollege㉿kali)-[~]
└─$ ./a.out
Array before sorting:64 25 12 22 11
Array after sorting:11 12 22 25 64
```

Program:

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

struct node {
    int data;
    struct node *next;
};

struct node *front = NULL;
struct node *rear = NULL;

void enqueue()
{
    int d;
    printf("Enter number to insert: ");
    scanf("%d", &d);

    struct node* new_n;
    new_n = (struct node*)malloc(sizeof(struct node));
    new_n->data = d;
    new_n->next = NULL;

    if (front == NULL && rear == NULL) {
        front = rear = new_n;
    } else {
        rear->next = new_n;
        rear = new_n;
    }
}

void dequeue()
{
    struct node *temp;

    if (front == NULL && rear == NULL) {
        printf("\nQueue is Empty\n");
    } else {
        temp = front;
        front = front->next;
        free(temp);

        if (front == NULL) {
            rear = NULL;
        }
    }
}

void display()
{
    struct node* temp;

    if (front == NULL && rear == NULL) {
        printf("\nQueue is Empty\n");
    } else {
        temp = front;
        while (temp) {
            printf(" %d ", temp->data);
            temp = temp->next;
        }
        printf("\n");
    }
}

int main() {
    int choice;

    while (1) {
        printf("\nSelect choice from the following:");
        printf("\n[1] Enqueue element into the queue");
        printf("\n[2] Dequeue element from the queue");
        printf("\n[3] Display elements in the queue");
        printf("\n[4] Exit");
        printf("\nEnter Choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                enqueue();
                break;

            case 2:
                dequeue();
                break;

            case 3:
                display();
                break;

            case 4:
                exit(0);

            default:
                printf("Invalid choice!!");
        }
    }

    return 0;
}
```

OUTPUT:-



```
┌──(yccollege㉿kali)-[~]
└─$ vi practical6.c

┌──(yccollege㉿kali)-[~]
└─$ gcc practical6.c

┌──(yccollege㉿kali)-[~]
└─$ ./a.out

Select choice from the following:
[1] Enqueue element into the queue
[2] Dequeue element from the queue
[3] Display elements in the queue
[4] Exit
Enter Choice: 1
Enter number to insert: 78

Select choice from the following:
[1] Enqueue element into the queue
[2] Dequeue element from the queue
[3] Display elements in the queue
[4] Exit
Enter Choice: 1
Enter number to insert: 23
```

```
Select choice from the following:
[1] Enqueue element into the queue
[2] Dequeue element from the queue
[3] Display elements in the queue
[4] Exit
Enter Choice: 1
Enter number to insert: 45

Select choice from the following:
[1] Enqueue element into the queue
[2] Dequeue element from the queue
[3] Display elements in the queue
[4] Exit
Enter Choice: 1
Enter number to insert: 67

Select choice from the following:
[1] Enqueue element into the queue
[2] Dequeue element from the queue
[3] Display elements in the queue
[4] Exit
Enter Choice: 2

Select choice from the following:
[1] Enqueue element into the queue
[2] Dequeue element from the queue
[3] Display elements in the queue
[4] Exit
Enter Choice: 3
 23  45  67



Select choice from the following:
[1] Enqueue element into the queue
[2] Dequeue element from the queue
[3] Display elements in the queue
[4] Exit
Enter Choice: 4
```