Name: Muhammad Zohaib khan

Reg# BSE203003

Subject: Data Structure Lab

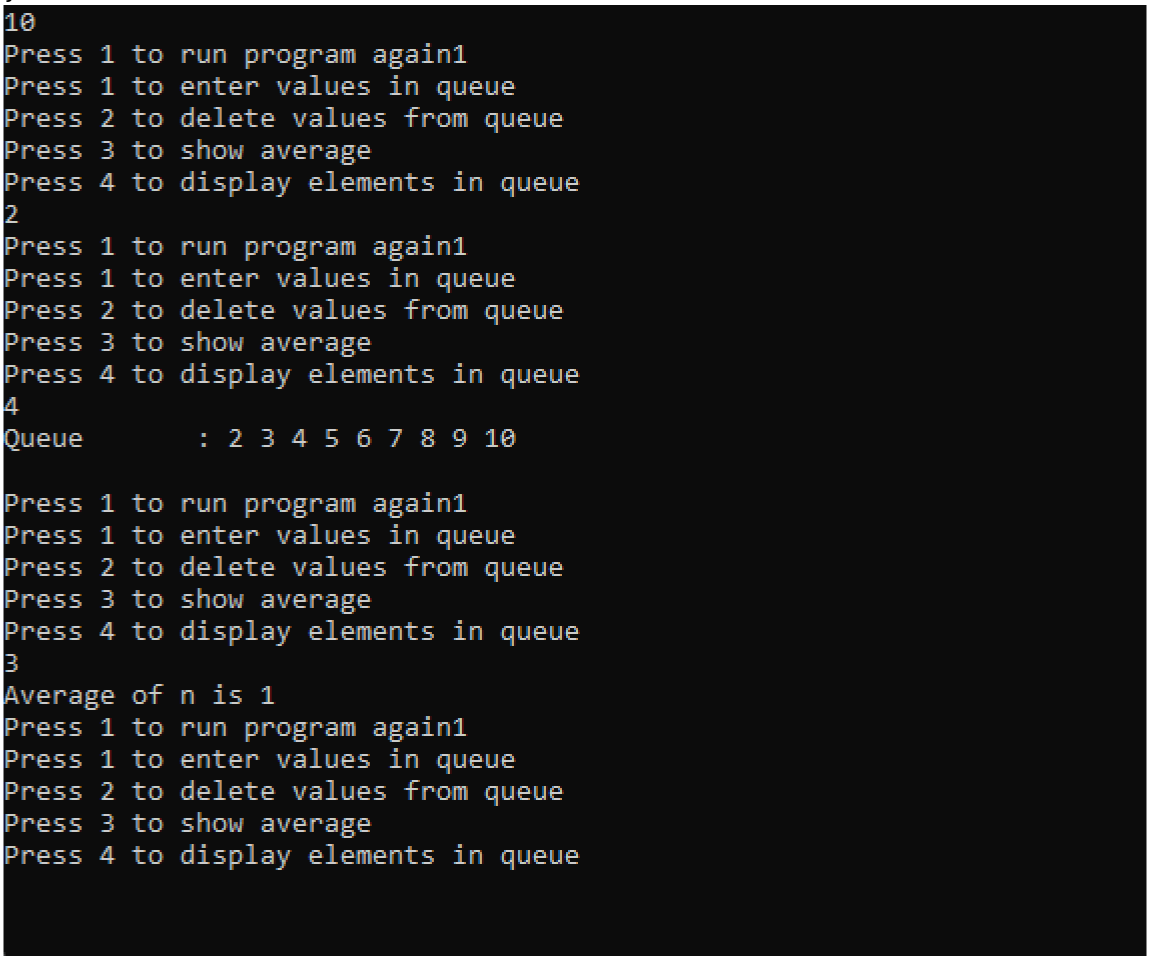
Section:1

Date:24-11-2021

**Practice Task 1:**

```
#include<iostream>
using namespace std;
#define size 10
class queue {
private:
	int q[size];
	int front;
	int rear;
	int sum;
	int n;
public:
	queue()
	{
		front = rear = -1;
		sum = 0;
		n = 0;

	}


	void enque(int v)
	{
		if (isfull())
		{
			cout << "Queue is  full" << endl;
		}
		else if (isempty())
		{
			front = rear = 0;
		}
		else
		{
			rear = (rear + 1) % size;
		}
		q[rear] = v;

	}
	void deque()
	{
		if (isempty())
		{
			cout << "Queue is empty" << endl;
		}
		else if (front == rear)
		{
			front = rear = -1;
		}
		else {
			sum = sum + q[front];
			n++;
			front = front + 1 % size;
		}
	}
	void avg()
	{
		int avg;
		avg = sum / n;
		cout << "Average of n is " << avg << endl;
	}
	void display()
	{
		int count = (rear + size - front) % size + 1;
		cout << "Queue       : ";
		for (int i = 0; i < count; i++)
		{
			int index = (front + i) % size;
			cout << q[index] << " ";
		}
		cout << "\n\n";
	}
	bool isempty()
	{
		if (front == -1 && rear == -1) {
			return true;
		}
		return false;
	}
	bool isfull()
	{
		if ((rear + 1) % size == front)
		{
			return true;
		}
		return false;
	}

};
int main()
{
	queue obj;
	int q;
	int option,ch;
	do {
		cout << "Press 1 to enter values in queue\n";
		cout << "Press 2 to delete values from queue\n";
		cout << "Press 3 to show average\n";
		cout << "Press 4 to display elements in queue\n";
		cin >> option;
		switch (option)
		{
		case 1:
			cout << "Enter values in Circuler queue : " << endl;
			for (int i = 0; i < 10; i++)
			{

				cin >> q;
				obj.enque(q);
			}break;
		case 2:
			obj.deque();
			break;
		case 3:
			obj.avg();
			break;
		case 4:
			obj.display();
			break;
		}cout << "Press 1 to run program again";
		cin >> ch;
	} while (ch == 1);
	system("pause");
```

```
}
```

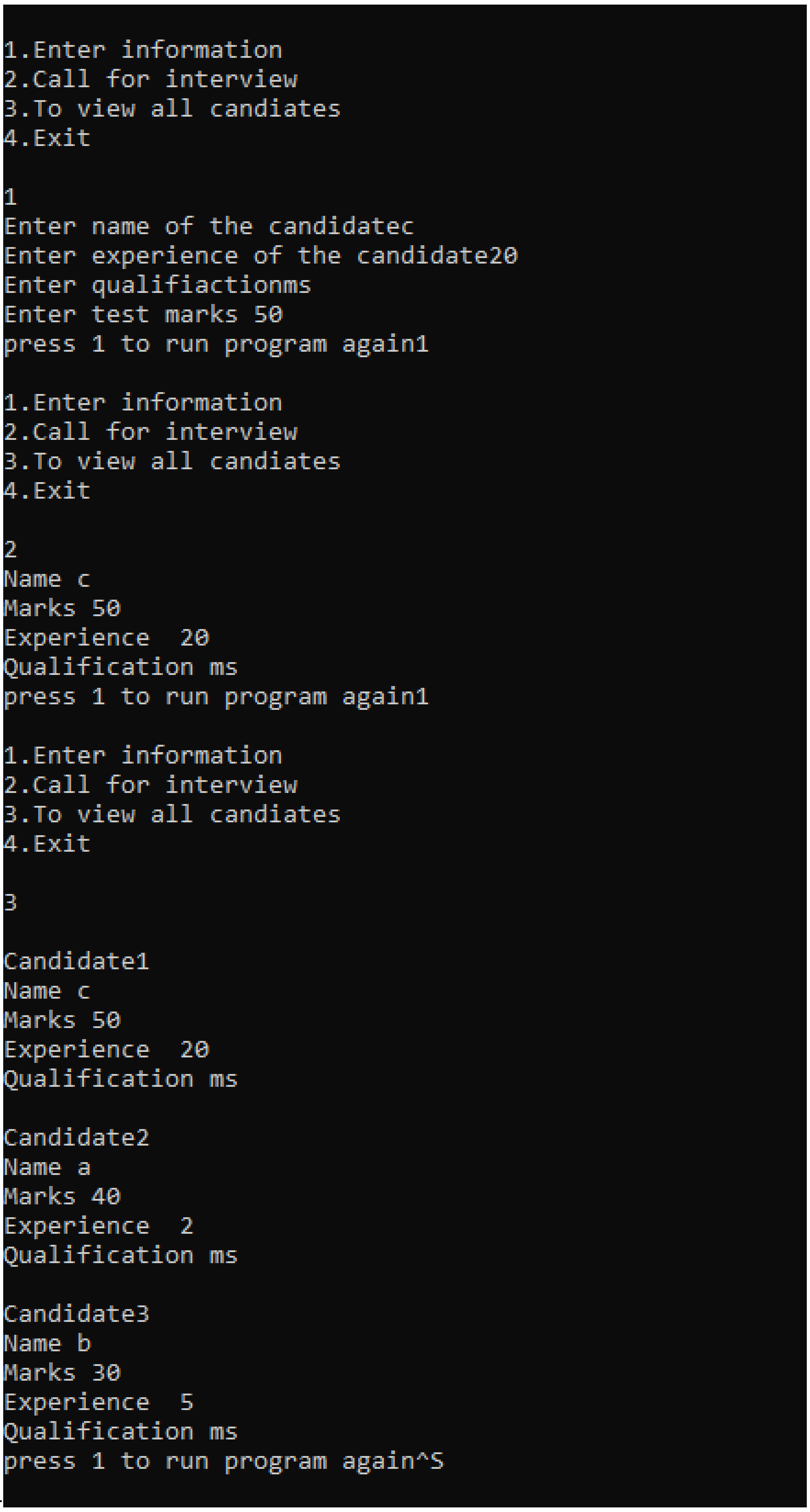


```
10
Press 1 to run program again1
Press 1 to enter values in queue
Press 2 to delete values from queue
Press 3 to show average
Press 4 to display elements in queue
2
Press 1 to run program again1
Press 1 to enter values in queue
Press 2 to delete values from queue
Press 3 to show average
Press 4 to display elements in queue
4
Queue       : 2 3 4 5 6 7 8 9 10

Press 1 to run program again1
Press 1 to enter values in queue
Press 2 to delete values from queue
Press 3 to show average
Press 4 to display elements in queue
3
Average of n is 1
Press 1 to run program again1
Press 1 to enter values in queue
Press 2 to delete values from queue
Press 3 to show average
Press 4 to display elements in queue
```

**Practice Task 2:**

```
#include<iostream>
#include<string>
using namespace std;
class candidates
{
	string name;
	int experience;
	string qualification;
	int test_marks;
public:
	void set_data()
	{
		cout << "Enter name of the candidate";
		cin >> name;
		cout << "Enter experience of the candidate";
		cin >> experience;
		cout << "Enter qualifiaction";
		cin >> qualification;
		cout << "Enter test marks ";
		cin >> test_marks;

	}
	string get_name()
	{
		return name;
	}
	int get_exp()
	{
```

```
                return experience;
        }
        string get_qualf()
        {
                return qualification;
        }
        int get_marks()
        {
                return test_marks;
        }
};
class p_queue {
        candidates c[10];
        int front, rear;
        const int max = 10;
public:
        p_queue()
        {
                front = rear = -1;
        }
        void enqueue()
        {
                if (rear == max - 1)
                {
                        cout << "Queue is full" << endl;
                }
                else
                {
                        if (front = -1)
                        {
                                front = 0;
                        }
                        candidates temp;
                        c[++rear].set_data();
                        for (int i = front; i <= rear;i++)
                        {

                                for (int j = i + 1; j <= rear; j++)
                                {
                                        if ( c[j].get_qualf() == "ms"&&c[i].get_qualf() !=
"ms")

                                        {
                                                temp = c[i];
                                                c[i] = c[j];
                                                c[j] = temp;
                                        }
                                        if (c[j].get_exp() > 3)
                                        {
                                                temp = c[i];
                                                c[i] = c[j];
                                                c[j] = temp;
                                        }
                                        if (c[j].get_marks() > c[i].get_marks())
                                        {
                                                temp = c[i];
                                                c[i] = c[j];
                                                c[j] = temp;
                                        }

                                }
                        }
                }
        }
        void display()
        {
                if (front == -1)
                {
                        cout << "It is empty" << endl;
                }
                else
                {
                        for (int k = 0; k <= rear; k++)
                        {
                                cout << "\nCandidate" << k + 1 << endl;
                                cout << "Name " << c[k].get_name() << endl;
                                cout << "Marks " << c[k].get_marks() << endl;
                                cout << "Experience  " << c[k].get_exp() << endl;
                                cout << "Qualification " << c[k].get_qualf() << endl;
                        }
                }
        }
        void display_interview()
        {
                if (front == -1)
                {
                        cout << "It is empty" << endl;
                }
                else
                {


                                cout << "Name " << c[front].get_name() << endl;
                                cout << "Marks " << c[front].get_marks() << endl;
                                cout << "Experience  " << c[front].get_exp() << endl;
                                cout << "Qualification " << c[front].get_qualf() << endl;

                }
        }
        void dequeaue()
        {
                if (front == -1)
                {
                        cout << "queue is empty";
                }
                else
                {
                        cout << "deleted candidatee" << c[front++].get_name() << endl;
                }
        }
};
int main()
{
        p_queue p;
        int option,ch;
        do {
                cout << "\n1.Enter information \n2.Call for interview \n3.To view all
candiates\n4.Exit\n" << endl;
                cin >> ch;
                switch (ch)
                {
                case 1:
                        p.enqueue();
                        break;
                case 2:
                        p.display_interview();
                        break;
                case 3:
                        p.display();
                        break;
                case 4:
                        exit(0);
                        break;

                }
                cout << "press 1 to run program again";
                cin >> option;
        } while (option == 1);
```



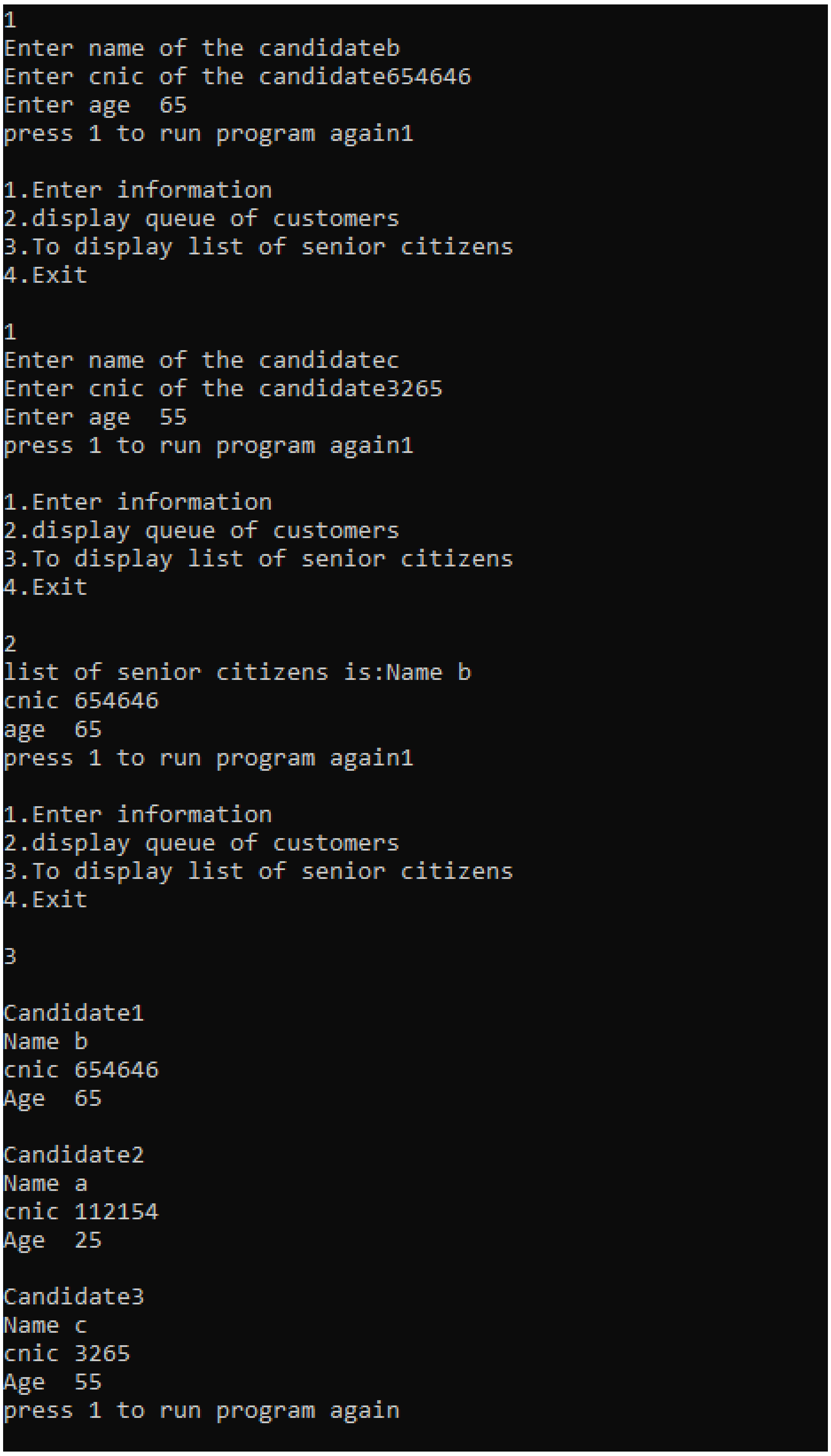
```
1.Enter information
2.Call for interview
3.To view all candiates
4.Exit

1
Enter name of the candidatec
Enter experience of the candidate20
Enter qualifiactionms
Enter test marks 50
press 1 to run program again1

1.Enter information
2.Call for interview
3.To view all candiates
4.Exit

2
Name c
Marks 50
Experience  20
Qualification ms
press 1 to run program again1

1.Enter information
2.Call for interview
3.To view all candiates
4.Exit

3

Candidate1
Name c
Marks 50
Experience  20
Qualification ms

Candidate2
Name a
Marks 40
Experience  2
Qualification ms

Candidate3
Name b
Marks 30
Experience  5
Qualification ms
press 1 to run program again^S
```

```
}
```

**Practice Task 3:**

```
#include<iostream>
#include<string>
using namespace std;
class customers
{
        string name;
        int cnic;
        int age;
public:
        void set_data()
        {
                cout << "Enter name of the candidate";
                cin >> name;
                cout << "Enter cnic of the candidate";
                cin >> cnic;

                cout << "Enter age  ";
                cin >> age;

        }
        string get_name()
        {
                return name;
        }
        int get_cnic()
        {
                return cnic;
        }

        int get_age()
        {
                return age;
        }
};
class bill_paying {
        customers c[10];
        int front, rear;
        const int max = 10;
public:
        bill_paying()
        {
                front = rear = -1;
        }
        void enqueue()
        {
                if (rear == max - 1)
                {
                        cout << "Queue is full" << endl;
                }
                else
                {
                        if (front = -1)
                        {
                                front = 0;
                        }
                        customers temp;
                        c[++rear].set_data();
                        for (int i = front; i <= rear;i++)
                        {

                                for (int j = i + 1; j <= rear; j++)
                                {
                                        if ( c[j].get_age() >=60)
                                        {
                                                temp = c[i];
                                                c[i] = c[j];
                                                c[j] = temp;
                                        }


                                }
                        }

                }
        }
        void display()
        {
                if (front == -1)
                {
                        cout << "It is empty" << endl;
                }
                else
                {
                        for (int k = 0; k <= rear; k++)
                        {
                                cout << "\nCandidate" << k + 1 << endl;
                                cout << "Name " << c[k].get_name() << endl;
                                cout << "cnic " << c[k].get_cnic() << endl;
                                cout << "Age  " << c[k].get_age() << endl;

                        }
                }
        }
        void display_senior()
        {
                if (front == -1)
                {
                        cout << "It is empty" << endl;
                }
                else
                {
                        cout << "list of senior citizens is:";
                        for (int k = 0; k <= rear; k++)

                        {
                                if (c[k].get_age() > 60) {
                                        cout << "Name " << c[k].get_name() << endl;
                                        cout << "cnic " << c[k].get_cnic() << endl;
                                        cout << "age  " << c[k].get_age() << endl;
                                }
                        }
                }
        }
        void dequeaue()
        {
                if (front == -1)
                {
                        cout << "queue is empty";
                }
                else
                {
                        cout << "deleted customer" << c[front++].get_name() << endl;
                }
        }
};
int main()
{
        bill_paying p;
        int option,ch;
```

```
	do {
		cout << "\n1.Enter information \n2.display queue of customers \n3.To 
display list of senior citizens\n4.Exit\n" << endl;
		cin >> ch;
		switch (ch)
		{
		case 1:
			p.enqueue();
			break;
		case 2:
			p.display_senior();
			break;
		case 3:
			p.display();
			break;
		case 4:
			exit(0);
			break;

		}
		cout << "press 1 to run program again";
		cin >> option;
	} while (option == 1);
}
```



```
1
Enter name of the candidateb
Enter cnic of the candidate654646
Enter age  65
press 1 to run program again1

1.Enter information
2.display queue of customers
3.To display list of senior citizens
4.Exit

1
Enter name of the candidatec
Enter cnic of the candidate3265
Enter age  55
press 1 to run program again1

1.Enter information
2.display queue of customers
3.To display list of senior citizens
4.Exit

2
list of senior citizens is:Name b
cnic 654646
age  65
press 1 to run program again1

1.Enter information
2.display queue of customers
3.To display list of senior citizens
4.Exit

3

Candidate1
Name b
cnic 654646
Age  65

Candidate2
Name a
cnic 112154
Age  25

Candidate3
Name c
cnic 3265
Age  55
press 1 to run program again
```