**// code for implementing queue using Array;**

package queue;

import java.util.Scanner;

public class impqueusingarr {

//Initalizing array of name queue

int size;

int queue[];

impqueusingarr(int size) {

this.size = size;

queue = new int[size];

}

public static int front = -1;

public static int rare = -1;

// Function to add element in queue

public void Enque(int data) {

if (rare == size - 1) {

System.out.println("Queue is full");

} else if (front == -1 && rare == -1) {

front = rare = 0;

queue[rare] = data;

} else {

rare++;

queue[rare] = data;

}

}

// Function to remove element from queue

public void deque() {

if (front == -1 && rare == -1) {

System.out.println("Queue is empty");

} else if (front == rare) {

front = rare = -1;

} else {

front++;

}

}

// Function to check whether queue is empty or not

public void isEmpty() {

if (front == -1 && rare == -1) {

System.out.println("Queue is empty");

} else {

System.out.println("Queue is Not empty");

}

}

// Function to display elements in queue

public void display() {

int temp = front;

if (front == -1 && rare == -1) {

System.out.println("Queue is empty");

} else {

while (temp < (rare + 1)) {

System.out.println(queue[temp]);

temp++;

}

}

}

// Function to count number of elements in queue

public void size() {

size = 0;

int temp = front;

if (front == -1 && rare == -1) {

System.out.println("Size is = " + size);

} else {

while (temp < (rare + 1)) {

size++;

temp++;

}

System.out.println("Size is = " + size);

}

}

// Function to peek first element of queue

public void peekfront() {

System.out.print("front element is = " + queue[front]);

}

// Function to peek last element of queue

public void peekrare() {

System.out.print("Rare element is = " + queue[rare]);

}

//Main function

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter size of array = ");

int size = sc.nextInt();

impqueusingarr arr = new impqueusingarr(size);

int ch;

// for a while loop

while (true) {

// List of choise

System.out.println("1.Enqueue");

System.out.println("2.Dequeue");

System.out.println("3.Is empty");

System.out.println("4.size");

System.out.println("5.Peek front");

System.out.println("6.Peek rare");

System.out.print("Enter youre choise = ");

ch = sc.nextInt();

switch (ch) {

case 1:

System.out.print("Enter the data that you want to insert = ");

int data = sc.nextInt();

arr.Enque(data);

break;

case 2:

arr.deque();

break;

case 3:

arr.isEmpty();

break;

case 4:

arr.size();

break;

case 5:

arr.peekfront();

System.out.println();

break;

case 6:

arr.peekrare();

System.out.println();

break;

case 0:

System.exit(0);

break;

}

}

} }

**7.Results:->**

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 2

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 4

Size is = 2

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 3

Queue is Not empty

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 5

front element is = 55

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 6

Rare element is = 7

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 0

Process finished with exit code 0

Enter size of array = 5

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 1

Enter the data that you want to insert = 18

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 1

Enter the data that you want to insert = 55

1.Enqueue

2.Dequeue

3.Is empty

4.size

5.Peek front

6.Peek rare

Enter youre choise = 1

Enter the data that you want to insert = 7