**// STACK**

import java.util.Scanner;

class Stack {

static Scanner ip = new Scanner(System.in);

//System.out.println("enter the size of array");

static int size = ip.nextInt();

static int [] a = new int [size];

static int top =-1, value;

public static void push(int data) {

if(top < size-1)

a[++top] = data;

else

System.out.println("Stack OVERFLOW");

}

public static void pop() {

top--;

if(top<0)

System.out.println("Stack EMPTY");

}

public static int top1() {

if(top>=0)

return a[top];

return 0;

}

public static void print() {

System.out.print("Stack:");

for(int i=0;i<=top;i++)

System.out.print(a[i]+" ");

System.out.println();

}

public static void main(String[] arg) {

System.out.println("the size of array is "+size);

push(4); print();

push(8); print();

push(7); print();

push(6); print();

//push(3); print();

value = top1(); System.out.println("top of the stack = "+value);

System.out.println("size of the stack = "+(top+1)+"\n");

pop(); print();

pop(); print();

value = top1(); System.out.println("top of the stack = "+value);

System.out.println("size of the stack = "+(top+1)+"\n");

pop(); print();

pop();

value = top1(); System.out.println("top of the stack = "+value);

System.out.println("size of the stack = "+(top+1)+"\n");

push(1); print();

push(0); print();

value = top1(); System.out.println("top of the stack = "+value);

System.out.println("size of the stack = "+(top+1)+"\n");

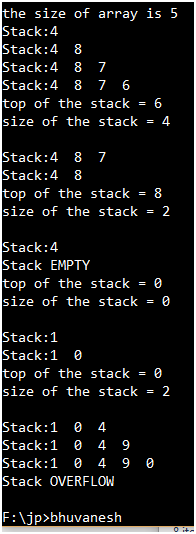
push(4); print();

push(9); print();

push(0); print();

push(9);

}



**// Queue**

import java.util.Scanner;

class Queue {

static Scanner ip = new Scanner(System.in);

//System.out.println("enter the n of array");

static int n = ip.nextInt();

static int [] a = new int [n];

static int front =-1, rear =-1, value;

public static void main(String[] arg) {

System.out.println("the size of array is "+n);

rear(4); print();

rear(8); print();

rear(7); print();

rear(6); print();

//rear(3); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

front(); print();

front(); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

front(); print();

front(); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

rear(1); print();

rear(0); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

rear(4); print();

rear(9); print();

rear(0); print();

rear(7);

}

public static void rear(int data) {

if((rear==-1)&&(front==-1))

front++;

if(rear < n-1)

a[++rear] = data;

else

System.out.println("Queue OVERFLOW");

}

public static void front() {

front++;

}

public static void Qsize() {

System.out.println("Queue size = "+(rear-front+1)+"\n");

if(front == (rear+1)%n) {

front =-1; rear =-1;

System.out.println("Queue EMPTY");

}

}

public static int firstInQ() {

if(front<=rear)

return a[front];

return 0;

}

public static void print() {

System.out.print("Queue:");

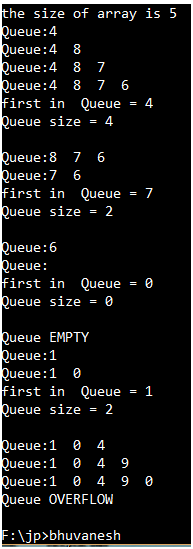
for(int i=front;i<=rear;i++)

System.out.print(a[i]+" ");

System.out.println();

}

}



**// CIRCULAR QUEUE**

import java.util.Scanner;

class CircularQ {

static Scanner ip = new Scanner(System.in);

//System.out.println("enter the size of array");

static int size = ip.nextInt();

static int [] a = new int [size];

static int front =1, rear =1, value;

public static void main(String[] arg) {

System.out.println("the size of array is "+size);

enQueue(4); print();

enQueue(8); print();

enQueue(7); print();

enQueue(1); print();

//enQueue(3); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

enQueue(10); print();

deQueue(); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

deQueue(); print();

enQueue(13); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

deQueue(); print();

deQueue(); print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

deQueue(); print();

deQueue(); //print();

value = firstInQ(); System.out.println("first in Queue = "+value);

Qsize();

enQueue(4); print();

enQueue(9); print();

enQueue(21); print();

//enQueue(0); print();

//enQueue(7);

}

public static void enQueue(int data) {

if((rear==1)&&(front==1))

front++;

if(rear < size-1)

a[++rear] = data;

else {

rear= (rear+1)%size; a[rear] = data; }

/\*else

System.out.println("Queue OVERFLOW");\*/

}

public static void deQueue() {

if(front<size-1)

front++;

else

front = (front+1)%size;

}

public static void Qsize() {

if(front == ((rear+1)%size)) {

//rear =-1; front =-1;

System.out.println("Queue EMPTY");

}

else if(rear<front)

System.out.println("Queue size = "+((size-front)+rear+1)+"\n");

else

System.out.println("Queue size = "+(rear-front+1)+"\n");

}

public static int firstInQ() {

if(front==(rear+1)%size)

return 0;

else

return a[front];

//return 0;

}

public static void print() {

if(rear<front) {

System.out.print("Queue:");

for(int i=0;i<=rear;i++)

System.out.print(a[i]+" ");

for(int i=front;i<=size-1;i++)

System.out.print(a[i]+" ");

System.out.println();

}

else {

System.out.print("Queue:");

for(int i=front;i<=rear;i++)

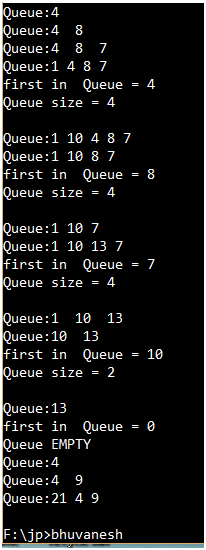
System.out.print(a[i]+" ");

System.out.println();

}

}

}



**// PRIORITY QUEUE**

import java.util.Scanner;

class PriorityQ {

static Scanner ip = new Scanner(System.in);

//System.out.println("enter the n of array");

static int n = ip.nextInt();

static int [] a = new int [n];

static int front =1, rear =1, value;

public static void enQueue(int data) {

if((rear==1)&&(front==1))

front++;

if(rear < n-1)

a[++rear] = data;

else

System.out.println("Queue OVERFLOW");

}

public static void selectHighPriority(int front, int rear) {

int high = front;

for(int p=front;p<=rear;p++) {

if(a[high]<a[p])

high=p;

}

if(high!=front) {

a[front] = a[high] + a[front];

a[high] = a[front] - a[high];

a[front] = a[front] - a[high];

}

}

public static void deQueue() {

System.out.println("bring the highest priority element to the front of the Queue");

selectHighPriority(front, rear); print();

System.out.println("highest priority got the service and left from the Q");

front++;

}

public static void print() {

System.out.print("Queue:");

for(int i=front;i<=rear;i++)

System.out.print(a[i]+" ");

System.out.println();

}

public static void main(String[] arg) {

enQueue(4); print();

enQueue(8); print();

enQueue(10); print();

enQueue(3); print();

enQueue(9); print();

enQueue(35); print();

enQueue(13); print();

deQueue(); print();

deQueue(); print();

deQueue(); print();

deQueue(); print();

deQueue(); print();

deQueue(); print();

}

}

