**Assignment 6 | 9th January 2021**

**Question 1**

**Solution 1**

#include <stdio.h>

#define size 5

void insertq(int[], int);

void deleteq(int[]);

void display(int[]);

int front = - 1;

int rear = - 1;

int main()

{

int n, ch;

int queue[size];

do

{

printf("\n\n Circular Queue:\n1. Insert \n2. Delete\n3. Display\n0. Exit");

printf("\nEnter Choice 0-3? : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("\nEnter number: ");

scanf("%d", &n);

insertq(queue, n);

break;

case 2:

deleteq(queue);

break;

case 3:

display(queue);

break;

}

}while (ch != 0);

}

void insertq(int queue[], int item)

{

if ((front == 0 && rear == size - 1) || (front == rear + 1))

{

printf("queue is full");

return;

}

else if (rear == - 1)

{

rear++;

front++;

}

else if (rear == size - 1 && front > 0)

{

rear = 0;

}

else

{

rear++;

}

queue[rear] = item;

}

void display(int queue[])

{

int i;

printf("\n");

if (front > rear)

{

for (i = front; i < size; i++)

{

printf("%d ", queue[i]);

}

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

printf("%d ", queue[i]);

}

else

{

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

printf("%d ", queue[i]);

}

}

void deleteq(int queue[])

{

if (front == - 1)

{

printf("Queue is empty ");

}

else if (front == rear)

{

printf("\n %d deleted", queue[front]);

front = - 1;

rear = - 1;

}

else

{

printf("\n %d deleted", queue[front]);

front++;

}

}

**Question 3**

**Solution 3**

using System;

using System.Collections;

public class MyStack

{

public Stack s;

public int minEle;

public MyStack()

{

s = new Stack();

}

public void getMin()

{

if (s.Count==0)

Console.WriteLine("Stack is empty");

else

Console.WriteLine("Minimum Element in the " +

" stack is: " + minEle);

}

public void Peek()

{

if (s.Count==0)

{

Console.WriteLine("Stack is empty ");

return;

}

int t =(int)s.Peek();

Console.Write("Top Most Element is: ");

if (t < minEle)

Console.WriteLine(minEle);

else

Console.WriteLine(t);

}

public void Pop()

{

if (s.Count==0)

{

Console.WriteLine("Stack is empty");

return;

}

Console.Write("Top Most Element Removed: ");

int t = (int)s.Pop();

if (t < minEle)

{

Console.WriteLine(minEle);

minEle = 2\*minEle - t;

}

else

Console.WriteLine(t);

}

public void Push(int x)

{

if (s.Count==0)

{

minEle = x;

s.Push(x);

Console.WriteLine("Number Inserted: " + x);

return;

}

if (x < minEle)

{

s.Push(2 \* x - minEle);

minEle = x;

}

else

s.Push(x);

Console.WriteLine("Number Inserted: " + x);

}

}

public class main

{

public static void Main(String []args)

{

MyStack s = new MyStack();

s.Push(3);

s.Push(5);

s.getMin();

s.Push(2);

s.Push(1);

s.getMin();

s.Pop();

s.getMin();

s.Pop();

s.Peek();

}

}