Assignment Of Day 6

Name : Gayatri Vaibhav Surve.

Gmail id : [gayatrisurve91@gmail.com](mailto:gayatrisurve91@gmail.com)

**Question 1**

Write a program implementing insert, delete and display operation of Circular Queue.

**Question 2**

A Barua number is a number which consists of only zeroes and ones and has only one 1. Barua

number will start with 1. Given numbers, find out the multiplication of the numbers. Note: The

input may contain one decimal number and all other Barua numbers. (Assume that each number

is very large and total number of values give is also very large)

Input 1: 100 10 12 1000

Output 1: 12000000

Input 2: 100 121 1000000000000000

Output 2: 12100000000000000000

Input 3: 10 100 1000

Output 3: 1000000

**Question 3**

Implement push, pop and find the minimum element in a stack in O(1) time complexity.

**Answers ::**

**Question 1**

#include<stdio.h>

# define MAX 5

int cqueue\_arr[MAX];

int front = -1;

int rear = -1;

void insert(int item)

{

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

{

printf("Queue Overflow n");

return;

}

if(front == -1)

{

front = 0;

rear = 0;

}

else

{

if(rear == MAX-1)

rear = 0;

else

rear = rear+1;

}

cqueue\_arr[rear] = item ;

}

void deletion()

{

if(front == -1)

{

printf("Queue Underflown");

return ;

}

printf("Element deleted from queue is : %dn",cqueue\_arr[front]);

if(front == rear)

{

front = -1;

rear=-1;

}

else

{

if(front == MAX-1)

front = 0;

else

front = front+1;

}

}

void display()

{

int front\_pos = front,rear\_pos = rear;

if(front == -1)

{

printf("Queue is emptyn");

return;

}

printf("Queue elements :n");

if( front\_pos <= rear\_pos )

while(front\_pos <= rear\_pos)

{

printf("%d ",cqueue\_arr[front\_pos]);

front\_pos++;

}

else

{

while(front\_pos <= MAX-1)

{

printf("%d ",cqueue\_arr[front\_pos])

front\_pos++;

}

front\_pos = 0;

while(front\_pos <= rear\_pos)

{

printf("%d ",cqueue\_arr[front\_pos]);

front\_pos++;

}

}

printf("n");

}

int main()

{

int choice,item;

do

{

printf("1.Insertn");

printf("2.Deleten");

printf("3.Displayn");

printf("4.Quitn");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1 :

printf("Input the element for insertion in queue : ");

scanf("%d", &item);

insert(item);

break;

case 2 :

deletion();

break;

case 3:

display();

break;

case 4:

break;

default:

printf("Wrong choicen");

}

}while(choice!=4);

return 0;

}

**Question 2**

#include <stdio.h>

int main() {

int i,n,a[100],num,zero=0;

printf("\n how many barua numbers u want to enter : ");

scanf("%d",&n);

printf("\n enter those numbers :: ");

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

{

scanf("%d",&a[i]);

}

int product =1;

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

{

num=a[i];

while(num%10==0)

{

num=num/10;

++zero;

}

product=product\*num;

}

printf("\n \n the multiplication of numbers is %d",product);

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

{

printf("0");

}

}

**Question 3**

#include <stdio.h>

#include<stdlib.h>

struct node {

int data;

struct node \*next;

};

struct node \*top=NULL;

void push(int n)

{

struct node \*new;

new=(struct node \*)malloc(sizeof(struct node));

new->data=n;

new->next=top;

top=new;

}

void display()

{

if(top==NULL)

{

printf("\nStack is empty\n");

return;

}

struct node \*temp;

temp=top;

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

}

void max\_element()

{

if(top==NULL)

{

printf("\nStack is empty\n");

return;

}

struct node \*temp;

temp=top;

int max\_ele=temp->data;

while(temp!=NULL)

{

if(max\_ele<temp->data)

max\_ele=temp->data;

temp=temp->next;

}

printf("\n\n The maximum element from the stack is %d",max\_ele);

}

void min\_element()

{

if(top==NULL)

{

printf("\nStack is empty\n");

return;

}

struct node \*temp;

temp=top;

int min\_ele=temp->data;

while(temp!=NULL)

{

if(min\_ele>temp->data)

min\_ele=temp->data;

temp=temp->next;

}

printf("\n\n The minimum element from the stack is %d",min\_ele);

}

void pop\_ele()

{

if(top==NULL)

{

printf("\nStack is empty\n");

return;

}

struct node \*temp;

temp=top;

printf("\n The %d element is popped from the stack\n",top->data);

top=top->next;

free(temp);

}

int main() {

int ch,num;

while(1)

{

printf("\n the options are as follows : \n 1. push() \n 2.display() \n 3.max\_element from stack \n 4.min\_element from stack \n5.pop() \n6.exit from code \n\n\n Enter the choise : ");

scanf("%d",&ch);

switch(ch){

case 1:printf("\n Enter new element ");

scanf("%d",&num);

push(num);

break;

case 2:printf("\n See the elements of the stack :\n");

display();

break;

case 3:max\_element();

break;

case 4:min\_element();

break;

case 5:pop\_ele();

break;

case 6:exit(1);

break;

default: printf("\n Please choose the correct option\n");

break;

}

}

}