Heaven's Light is Our Guide

Computer Science & Engineering
Rajshahi University of Engineering & Technology

Lab Manual

Module- 07

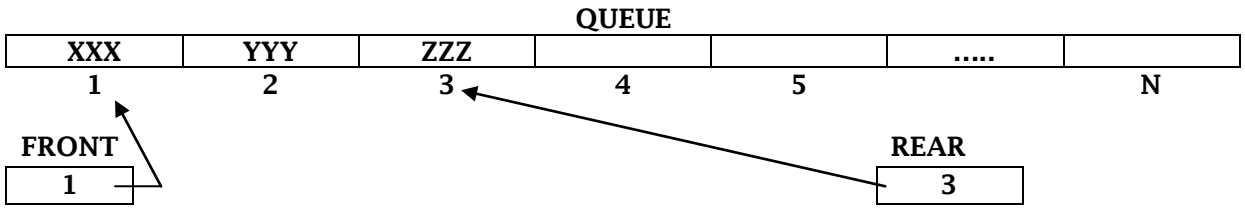
Course Title: Sessional based on CSE 1201
Course No. : CSE 1202

Experiment No. 7

Name of the Experiment: Queue

Duration: 1 cycle

Background Study: Chapter 6 (Theory and Problems of Data Structures Written by Seymour Lipschutz)



Insert an element: REAR: = REAR+1 (IF REAR>N → REAR: =1)
Delete an element: FRONT: = FRONT+1 (IF FRONT>N → FRONT: =1)
Queue Contains only one element: FRONT=REAR≠NULL
Queue is empty: FRONT: =NULL and REAR:=NULL

Problem I: Add an item into a Queue (PUSH).
Algorithm6.1: QINSERT (QUEUE, N, FRONT, REAR, ITEM)
This procedure inserts an ITEM into a queue.

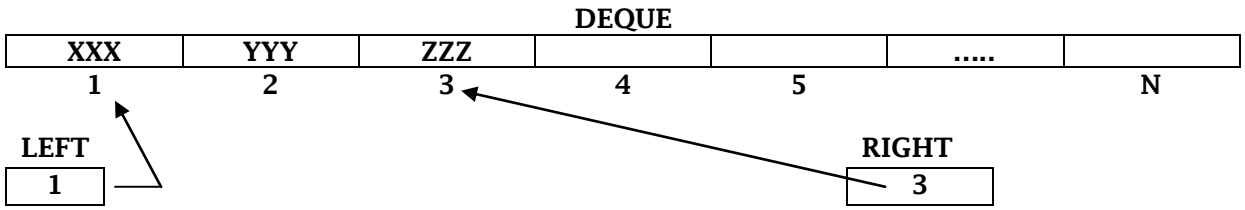
1. [Queue already filled]
   IF FRONT = 1 and REAR = N, or If FRONT = REAR + 1, then:
   Write: OVERFLOW, and Return.
2. IF FRONT=NULL, then: [Queue initially empty]
   Set FRONT:= 1 and REAR :=1.
   Else if REAR = N, then:
   Set REAR:=1.
   Else:
   Set REAR := REAR + 1.
   [End of If statement]
3. Set QUEUE[REAR]:= ITEM.
4. Return.

Flow Chart: Draw a flow chart.

Problem II: Delete an item from a queue (POP).
Algorithm6.2: QDELETE (QUEUE, N, FRONT, REAR, ITEM)
This procedure deletes an element from a queue and assigns it to the variable ITEM.

1. [Queue already Empty]
   IF FRONT = NULL then: Write: UNDERFLOW, and Return.
2. Set ITEM:= QUEUE[FRONT]
3. IF FRONT=REAR, then: [Queue contains only one element]
   Set FRONT:= NULL and REAR :=NULL.
   Else if FRONT = N, then:
   Set FRONT:=1.
   Else:
   Set FRONT := FRONT + 1.
   [End of If statement]
4. Return.

Flow Chart: Draw a flow chart.



LEFT = 1 and RIGHT = 1 or LEFT=RIGHT+1, Overflow
INSERT at right end: RIGHT:=RIGHT+1 (IF RIGHT>N → RIGHT: =1)
INSERT at left end: LEFT:=LEFT-1 (IF LEFT<1 → LEFT:=N)
DELETE at right end: RIGHT:=RIGHT-1(IF RIGHT<1 → RIGHT:=N)
DELETE ar left end: LEFT:=LEFT+1(IF LEFT>N → LEFT:=1)
LEFT=NULL and RIGHT=NULL: Underflow
LEFT=RIGHT≠NULL: Deque contains only one element.

Problem III: Insert an item into a deque at right end (rear end).
Algorithm6.3: DEQINSR (DEQUE, N, LEFT, RIGHT, ITEM)
This procedure inserts an ITEM into a deque at right end.

1. [Deque already filled]
   IF LEFT = 1 and RIGHT = N, or If LEFT = RIGHT + 1, then:
   Write: OVERFLOW, and Return.
2. IF RIGHT=NULL, then: [Deque initially empty]
   Set LEFT:= 1 and RIGHT :=1.
   Else if RIGHT = N, then:
   Set RIGHT:=1.
   Else:
   Set RIGHT := RIGHT + 1.
   [End of If statement]
3. Set DEQUE[RIGHT]:= ITEM.
4. Return.

Flow Chart: Draw a flow chart.

Problem IV: Insert an item into a deque at left end (front end).
Algorithm6.4: DEQINSL (DEQUE, N, LEFT, RIGHT, ITEM)
This procedure inserts an ITEM into a deque at left end.

1. [Deque already filled]
   IF LEFT = 1 and RIGHT = N, or If LEFT = RIGHT + 1, then:
   Write: OVERFLOW, and Return.
2. IF LEFT=NULL, then: [Deque initially empty]
   Set LEFT:= 1 and RIGHT :=1.
   Else if LEFT = 1, then:
   Set LEFT:=N.
   Else:
   Set LEFT := LEFT - 1.
   [End of If statement]
3. Set DEQUE[LEFT]:= ITEM.
4. Return.

Flow Chart: Draw a flow chart.

Problem V: Delete an item from a deque at right end (rear end).
Algorithm6.6: DEQDELR (DEQUE, N, LEFT, RIGHT, ITEM)
This procedure deletes an ITEM from a deque at right end and assigns it to the variable ITEM.

1. [Deque already Empty]
   IF RIGHT = NULL then: Write: UNDERFLOW, and Return.
2. Set ITEM:= DEQUE[RIGHT]

3. IF RIGHT=LEFT, then: [Deque contains only one element]
   Set LEFT:= NULL and RIGHT :=NULL.
   Else if RIGHT = 1, then:
   Set RIGHT:=N.
   Else:
   Set RIGHT := RIGHT - 1.
   [End of If statement]
4. Return.

Flow Chart: Draw a flow chart.

Problem VI: Delete an item from a deque at left end (front end).
Algorithm6.6: DEQDELL (DEQUE, N, LEFT, RIGHT, ITEM)
This procedure deletes an ITEM from a deque at left end and assigns it to the variable ITEM.

1. [Dequeue already Empty]
   IF LEFT = NULL then: Write: UNDERFLOW, and Return.
2. Set ITEM:= DEQUE[LEFT]
3. IF RIGHT=LEFT, then: [Dequeue contains only one element]
   Set LEFT:= NULL and RIGHT :=NULL.
   Else if LEFT = N, then:
   Set LEFT:=1.
   Else:
   Set LEFT := LEFT + 1.
   [End of If statement]
4. Return.

Flow Chart: Draw a flow chart.

Exercise: (**Priority Queue)**

[1] Delete first element from priority queue
[2] Insert an element into priority queue
[3] Delete first element from priority queue (array representation)
[4] Insert an element into priority queue (array representation)

MORE PROBLEMS

1. Programming Problems of Chapter 6 of “Data Structures” by Seymour Lipschutz.

LAB REPORT: You have to submit all assigned problems in next lab.

QUEUE

```
#include<stdio.h>
#define SIZE 6

void print(int queue[]){
  int i;
  printf("\nqueue: ");
  for(i=1;i<SIZE;i++)
    printf("%d ",queue[i]);
  printf("\n");
}

void QInsert(int Queue[],int *N,int *F,int *R,int *ITEM){
    if((*F==1 && *R==*N) || (*F==*R+1)){
        printf("\n------\n|Overflow|\n------");
        return;
    }
    if(*R==0)
        *R=1,*F=1;
    else if(*R==*N)
        *R=1;
    else
       *R = *R+1;
    Queue[*R] = *ITEM;
}
void QDelete(int Queue[], int *N,int *F,int *R,int *ITEM){
   if(*F==0){
        printf("\n------\n|Underflow|\n------");
        return;
   }
   *ITEM = Queue[*F];
   Queue[*F] = 0; /*Set Zero*/
   if(*R==*F)
       *R=0,*F=0;
   else if(*F==*N)
        *F=1;
   else
      *F=*F+1;
}
int main(){
  int op;
  int F=0,R=0,Queue[SIZE]={},N=SIZE-1;
  int ITEM=0;

  printf("\n1. Insert an element into Queue\n");
  printf("\n2. Delete an element from Queue\n");
  printf("\n3.Exit\n");

  while(1){
    printf("\n Enter Option: ");
    scanf("%d",&op);
    if(op ==1){
       printf("\n Enter Value: ");
       scanf("%d",&ITEM);
```

```
        QInsert(Queue, &N, &F, &R, &ITEM);
        printf("\nF = %d R = %d\n",F,R);
        print(Queue);
    }
    else if(op ==2){
        QDelete(Queue, &N, &F, &R, &ITEM);
        printf("\nF = %d R = %d\n",F,R);
        print(Queue);
    }
    else
      break;
  }
  return 0;
}
```

DEQUE

```
#include<stdio.h>
#define SIZE 6

void print(int deque[]){
  int i;
  printf("\nDeque: ");
  for(i=1;i<SIZE;i++)
    printf("%d ",deque[i]);
  printf("\n");
}

void deqInsR(int deque[], int *N,int *L,int *R,int *ITEM){
    if((*L==1 && *R==*N) || (*L==*R+1)){
        printf("\n------\n|Overflow|\n------");
        return;
    }
    if(*R==0)
        *R=1,*L=1;
    else if(*R==*N)
        *R=1;
    else
       *R = *R+1;
    deque[*R] = *ITEM;
}
void deqInsL(int deque[], int *N,int *L,int *R,int *ITEM){
    if((*L==1 && *R==*N) || (*L==*R+1)){
        printf("\n------\n|Overflow|\n------");
        return;
    }
    if(*L==0)
        *R=1,*L=1;
    else if(*L==1)
        *L=*N;
    else
       *L = *L-1;
    deque[*L] = *ITEM;
}
```

```
void deqDelR(int deque[], int *N,int *L,int *R,int *ITEM){
   if(*R==0){
        printf("\n------\n|Underflow|\n------");
        return;
   }
   *ITEM = deque[*R];
   deque[*R] = 0; /*Set Zero*/
   if(*R==*L)
       *R=0,*L=0;
   else if(*R==1)
        *R=*N;
   else
      *R=*R-1;
}
void deqDelL(int deque[], int *N,int *L,int *R,int *ITEM){
   if(*L==0){
        printf("\n------\n|Underflow|\n------");
        return;
   }
   *ITEM = deque[*L];
   deque[*L] = 0; /*Set Zero*/
   if(*R==*L)
       *R=0,*L=0;
   else if(*L==*N)
        *L=1;
   else
      *L=*L+1;
}
int main(){
  int op;
  int R=0,L=0,deque[SIZE]={},N=SIZE-1;
  int ITEM=0;

  printf("\n1.Insert an element into deque at right end(rear end)\n");
  printf("\n2. Insert an element into deque at left end(front end)\n");
  printf("\n3. Delete an element from deque at right end(rear end)\n");
  printf("\n4. Delete an element from deque at left end(front end)\n");
  printf("\n5.Exit\n");

  while(1){
    printf("\n Enter Option: ");
    scanf("%d",&op);
    if(op ==1){
       printf("\n Enter Value: ");
       scanf("%d",&ITEM);
       deqInsR(deque, &N, &L, &R, &ITEM);
       printf("\nL = %d R = %d\n",L,R);
       print(deque);
    }
    else if(op ==2){
       printf("\n Enter Value: ");
       scanf("%d",&ITEM);
       deqInsL(deque, &N, &L, &R, &ITEM);
       printf("\nL = %d R = %d\n",L,R);
       print(deque);
```

```
    }
    else if(op ==3){
       deqDelR(deque, &N, &L, &R, &ITEM);
       printf("\nL = %d R = %d\n",L,R);
       print(deque);
    }
    else if(op ==4){
       deqDelL(deque, &N, &L, &R, &ITEM);
       printf("\nL = %d R = %d\n",L,R);
       print(deque);
    }
    else
      break;
  }
  return 0;
}
```