Rajshahi University of Engineering & Technology

Department of Computer Science of Engineering

**Experiment No**: 07

**Name of Experiment**: Queue

|  |
| --- |
| Submitted to:  Rizoan Toufiq  Assistant Professor  Department of Computer Science & Engineering  Rajshahi University of Engineering & Technology |

|  |
| --- |
| Submitted by:  Name: Md. Ariful Islam  Roll No.: 1803046  Group: 2nd Thirty  Date of Exp.: 09-11-2019  Date of sub. : 17-11-2019  Series: 18 Series |

Machine configuration:

ASUS X510UF

CORE I5 8th Gen Processor

Up To 3.4 GHz

8 GB RAM

OS WIN 10

INDEX

|  |  |  |
| --- | --- | --- |
| Preblem No. | Name of Problem | Page No. |
| 01. | Inserting an item into a deque at right end | 04-05 |
| 02. | Inserting an item into a deque at left end | 06-07 |
| 03. | Deleting an item into a deque at right end | 08-09 |
| 04. | Deleting an item into a deque at left end | 10-11 |
| 05. | <empty> |  |
| 06. | <empty> |  |
| 07. | <empty> |  |
| 08 | <empty> |  |
| 09 | <empty> |  |
| 10. | <empty> |  |

# Theory: A Queue is a linear structure in which items may be added one by one only at the end or removed one by one only at the beginning.It means that the last item to be added to a queue is the last item to be removed.

The main concept of queue is First In - First Out.

# In a queue there is a pointer **REAR** that shows the **location** of **last data item.** There are two operations that are normally performed on any queue :

1. PUSH : Inserting an element into a queue.
2. POP : Deleting an element from a queue.

A Deque is a linear list in which elements can be added or removed at either end but not in the middle. The term deque is a constraction of the name double-ended queue.

In a deque there is two pointer LEFT and RIGHT which points toward the last item at left side and right side.

Problem No: 01

Problem Statement: Inserting an item into a Deque at right end.

Flow Chart :

START

AVAIL : NULL

!= = =

NEW: AVAIL

AVAIL: LINK[AVAIL]

LINK[NEW]: NULL

OVERFLOW

INFO[NEW] : ITEM

LINK[RIGHT]: NEW

RIGHT: NEW

END

# **Algorithm:** **DEQINSR(INFO,LINK,LEFT,RIGHT,AVAIL, ITEM)**

# This procedure inserts an ITEM into a Deque at left end.

# IF AVAIL:= NULL, then Write: OVERFLOW, and Return.

# [Deque already filled]

# Else Set NEW:= AVAIL and AVAIL:= LINK[AVAIL]

# and LINK[NEW]:= NULL.

# Set INFO[NEW]:= ITEM and LINK[RIGHT]:=NEW

# and RIGHT:= NEW.

# 4. Return.

Code:

|  |
| --- |
| #include<stdio.h>  int main(){  int info[10]={10,20,30,40,50,0,0,0,0,0};  int link[10]={2,3,4,5,0,7,8,9,10,0};  int left=1,right=5,avail=6,ptr,New,item;  if(avail==0)  printf("Overflow\n");  else{  New=avail;  avail=link[avail-1];  link[New-1]=0;  scanf("%d",&item);  info[New-1]=item;  link[right-1]=New;  right=New;  }  return 0;  } |

Problem No: 02

Problem Statement: Inserting an item into a Deque at left end.

Flow Chart :

START

AVAIL : NULL

!= = =

NEW: AVAIL

AVAIL: LINK[AVAIL]

LINK[NEW]: NULL

OVERFLOW

INFO[NEW] : ITEM

LINK[NEW]: LEFT

LEFT: NEW

END

# **Algorithm:** **DEQINSR(INFO,LINK,LEFT,RIGHT,AVAIL, ITEM)**

# This procedure insert an ITEM into a Deque at left end.

# IF AVAIL:= NULL, then Write: OVERFLOW, and Return.

# [Deque already filled]

# Else Set NEW:= AVAIL and AVAIL:= LINK[AVAIL]

# and LINK[NEW]:= NULL.

# Set INFO[NEW]:= ITEM and LINK[NEW]:=LEFT

# and LEFT:= NEW.

# 4. Return.

Code:

|  |
| --- |
| #include<stdio.h>  int main(){  int info[10]={10,20,30,40,50,0,0,0,0,0};  int link[10]={2,3,4,5,0,7,8,9,10,0};  int left=1,right=5,avail=6,ptr,New,item;  if(avail==0)  printf("Overflow\n");  else{  New=avail;  avail=link[avail-1];  link[New-1]=0;  scanf("%d",&item);  info[New-1]=item;  link[New]=left;  left=New;  }  return 0;  } |

Problem No: 03

Problem Statement: Deleting an item from a Deque at right end.

Flow Chart :

START

RIGHT : NULL

!= = =

ITEM: INFO[RIGHT]

LINK[RIGHT]: AVAIL

AVAIL: RIGHT

UNDERFLOW

RIGHT: RIGHT-1

LINK[RIGHT]: NULL

END

# **Algorithm:** **DEQDELR(INFO,LINK,LEFT,RIGHT,AVAIL, ITEM)**

# This procedure deletes an ITEM from a Deque at right end.

# IF RIGHT:= NULL, then Write: UNDERFLOW, and Return.

# [Deque already empty]

# Else Set ITEM:= INFO[RIGHT] and LINK[RIGHT]:= AVAIL

# and AVAIL:= RIGHT.

# Set RIGHT:= RIGHT-1 and LINK[RIGHT]:= NULL.

# 4. Return.

Code:

|  |
| --- |
| #include<stdio.h>  int main(){  int info[10]={10,20,30,40,50,0,0,0,0,0};  int link[10]={2,3,4,5,0,7,8,9,10,0};  int left=1,right=5,avail=6,ptr,New,item;    if(right==0)  printf("Underflow\n");  else{  item=info[right-1];  link[right-1]=avail;  avail=right;  right=right-1;  link[right]=0;  }  return 0;  } |

Problem No: 04

Problem Statement: Deleting an item from a Deque at left end.

Flow Chart :

START

LEFT : NULL

!= = =

ITEM: INFO[LEFT]

NEW: LINK[LEFT]

LINK[LEFT]: AVAIL

UNDERFLOW

AVAIL: LEFT

LEFT: NEW

END

# **Algorithm:** **DEQDELL(INFO,LINK,LEFT,RIGHT,AVAIL, ITEM)**

# This procedure deletes an ITEM from a Deque at left end.

# IF LEFT:= NULL, then Write: UNDERFLOW, and Return.

# [Deque already empty]

# Else Set ITEM:= INFO[LEFT] and NEW:= LINK[LEFT]

# and LINK[LEFT]:= AVAIL.

# Set AVAIL:= LEFT and LEFT:= NEW.

# 4. Return.

Code:

|  |
| --- |
| #include<stdio.h>  int main(){  int info[10]={10,20,30,40,50,0,0,0,0,0};  int link[10]={2,3,4,5,0,7,8,9,10,0};  int left=1,right=5,avail=6,ptr,New,item;    if(left==0)  printf("Underflow\n");  else{  item=info[left-1];  New=link[left-1];  Link[left-1]=avail;  Avail=left;  left=New;  }  return 0;  } |

Conclusion: I have completed all the problem with one-way link list.

But I’m feceing some problem in problem 3.

# THE END #