

- g. WAP to simulate the working of a queue of integers using an array. Provide the following operations:
- insert, delete, display. The program should print appropriate message for queue empty and queue overflow conditions.

Pseudo code

1. Initialize queue array [N], front = -1, rear = -1
2. function enqueue(x):
 - if queue full: print "overflow"
 - else if empty: front = rear = 0, queue[rear] = x
 - else: rear++, queue[rear] = x
3. function dequeue():
 - if empty: print "underflow"
 - else if one element: print and set front = rear = -1
 - else: print and front++
4. function peek():
 - if empty: print "queue empty"
 - else print queue elements from front to rear
5. main:
 - enqueue(1), enqueue(1), enqueue(1)
 - display
 - peek()
 - dequeue()
 - display()
 - enqueue(1), enqueue(1) // overflow

#

#include <stdio.h>
#include <stdlib.h>

#define N 5
int queue[N];
int front = -1;
int rear = -1;

void enqueue(int x) {

if (rear == N-1) {

printf("In queue overflow: %d", x);

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

front = rear = 0;

queue[rear] = x;

printf(" %d inserted into the",

queue. In %d, %d",

else {

rear++;

queue[rear] = x;

printf(" %d inserted into the queue. In %d, %d",

rear, x);

}

void dequeue() {

{

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

printf("In queue underflow: queue

is empty. In %d",

}

else if (front == rear) {

printf(" %d deleted from the queue. In %d",

queue[front]);

front = rear = -1;

}

else {
 printf (" %d deleted from the queue.\n",
 front++);

}
word display () {
 printf;

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

{
 printf ("in queue is empty.\n");

}
else {

 printf ("in queue elements are:");

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

{

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

 printf ("\n");

}

word peek () {

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

 printf ("in queue is empty.\n");

 }

 else {
 printf ("in front element is: %d\n", queue[front]);

 }

int main () {

 int choice, value;
 while (1)

{

 printf ("in ---- queue menu ----\n");
 printf ("1. enqueue (insert)\n");
 printf ("2. dequeue (delete)\n");
 printf ("3. peek (front element)\n");
 printf ("4. display queue\n");
 printf ("5. exit\n");
 printf ("enter your choice:");
 scanf ("%d", &choice);

switch (choice) {

case 1:

 printf ("Enter value to insert:");

 scanf ("%d", &value);

 enqueue (value);

 break;

case 2:

 dequeue ();

 break;

case 3:

 peek ();

 break;

case 4:

 display ();

 break;

case 5:

 printf ("in exiting program...");

 printf ("bye\n");

 exit (0);

default:

 printf ("in invalid choice! please try again.\n");

 }

}

return 0;

}

output

queue menu - - -

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

enter your choice: 2
queue underflow; queue is empty.

- - - queue menu - - -

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

enter your choice: 4
queue is empty.

- - - queue menu - - -

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

enter your choice: 1

enter value to insert: 10

10 inserted into the queue.

- - - queue menu - - -

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

enter your choice: 1.

enter value to insert: 11

11 inserted into the queue.

- - - queue menu - - -

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

enter your choice: 1

enter value to insert: 12

12 inserted into the queue.

- - - queue menu - - -

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

enter your choice: 1

enter value to insert: 13

13 inserted into the queue.

- - - queue menu - - -

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)

4. display queue.

5. exit.

Enter your choice: 1
Enter value to insert: 14.
14 inserted into the queue.

--- Queue menu ---

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

Enter your choice: 1
Enter value to insert: 15
Queue overflow!

--- Queue menu ---

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

Enter your choice: 4.
Queue elements are: 10 11 12 13 14

--- Queue menu ---

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

Enter your choice: 2.
10 deleted from the queue.

classmate

Date

Page

--- Queue menu ---

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

Enter your choice: 4
Queue elements are: 11 12 13 14

--- Queue menu ---

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

Enter your choice: 3
front element is: 11.

~~--- Queue menu ---~~

1. enqueue (insert)
2. dequeue (delete)
3. peek (front element)
4. display queue
5. exit

Enter your choice: 5
Exiting program ... good bye!

19/10/25

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 2

Queue Underflow! Queue is empty.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 4

Queue is empty.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 1

Enter value to insert: 10

10 inserted into the queue.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 1

Enter value to insert: 11

11 inserted into the queue.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 1

Enter value to insert: 12

12 inserted into the queue.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 1

Enter value to insert: 13

13 inserted into the queue.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 1

Enter value to insert: 14

14 inserted into the queue.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 1

Enter value to insert: 15

Queue Overflow!

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 4

Queue elements are: 10 11 12 13 14

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 2

10 deleted from the queue.

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 4

Queue elements are: 11 12 13 14

===== QUEUE MENU =====

1. Enqueue (Insert)
2. Dequeue (Delete)
3. Peek (Front Element)
4. Display Queue
5. Exit

Enter your choice: 3

Front element is: 11

===== QUEUE MENU =====

- 3. Peek (Front Element)
- 4. Display Queue
- 5. Exit

Enter your choice: 4

Queue elements are: 11 12 13 14

===== QUEUE MENU =====

- 1. Enqueue (Insert)
- 2. Dequeue (Delete)
- 3. Peek (Front Element)
- 4. Display Queue
- 5. Exit

Enter your choice: 3

Front element is: 11

===== QUEUE MENU =====

- 1. Enqueue (Insert)
- 2. Dequeue (Delete)
- 3. Peek (Front Element)
- 4. Display Queue
- 5. Exit

Enter your choice: 5

Exiting program... Goodbye!

Process returned 0 (0x0) execution time : 72.935 s
Press any key to continue.

|