|  |
| --- |
| /\*Description: Array implementation of Circular Queue |
|  | \* Learner: ARSHEE QURESHI |
|  | \* created on: 10th AUGUST 2017 |
|  | \*/ |
|  |  |
|  | #include<stdio.h> |
|  | #include<math.h> |
|  | #include<stdlib.h> |
|  | #define MAXSIZE 10 |
|  |  |
|  |  |
|  |  |
|  | int queue[MAXSIZE], front=-1, rear=-1; |
|  | int isFull(){ |
|  | //return 1 if stack is full else returns -1 |
|  |  |
|  | return ((rear==MAXSIZE-1 && front==0)||(rear+1==front)); |
|  |  |
|  | } |
|  |  |
|  | int isEmpty(){ |
|  | //return 1 if stack is empty else returns-1 |
|  | if(front==-1&&rear==-1) |
|  | return 1; |
|  | else |
|  | return 0; |
|  |  |
|  | } |
|  |  |
|  | int peek(){ |
|  | //return elements at the top of stack |
|  | return queue[front]; |
|  | } |
|  |  |
|  | void insertQueue(int e) |
|  | { |
|  | if(!isFull()) |
|  | { |
|  | if(isEmpty()) |
|  | { |
|  | front++; |
|  | } |
|  | queue[(rear++)%MAXSIZE]=e; |
|  | printf("Your element is successfully inserted\n"); |
|  | } |
|  | else |
|  | { |
|  | printf("Queue is full \n Element cannot be inserted"); |
|  | } |
|  |  |
|  |  |
|  | } |
|  | void deleteQueue() |
|  | { |
|  | if(!isEmpty()) |
|  | { |
|  | int d=queue[front]; |
|  | if(front==rear) |
|  | { |
|  | front=rear=-1; |
|  | } |
|  | else |
|  | { |
|  | front=(front+1)%MAXSIZE; |
|  | } |
|  | printf("Element deleted is %d\n",d); |
|  | } |
|  | } |
|  |  |
|  | void display() |
|  | { |
|  | //return element at the front of Queue |
|  | int i=front; |
|  | if(!isEmpty()) |
|  | { |
|  | while(i!=rear) |
|  | { |
|  | printf("%d\t",queue[i]); |
|  | i=(i+1)%MAXSIZE; |
|  | } |
|  | printf("%d\t",queue[i]); |
|  | } |
|  | else |
|  | printf("\nQueue is Empty."); |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  | int main(){ |
|  | int choice,e; |
|  | do |
|  | { |
|  | printf("\nDisplay Menu\n1.Peek\n2.Insert\n3.Delete\n4.Display\n5.Exit\n"); |
|  | printf("Enter choice\n"); |
|  | scanf("%d",&choice); |
|  | switch(choice){ |
|  | case 1: |
|  |  |
|  | printf("Element at front of queue : %d\n",peek()); |
|  |  |
|  | break; |
|  | case 2: |
|  | printf("Enter element to be inserted\n "); |
|  | scanf("%d",&e); |
|  | insertQueue(e); |
|  | break; |
|  | case 3:deleteQueue(); |
|  | break; |
|  | case 4:display(); |
|  | break; |
|  | case 5: |
|  | exit (0); |
|  | break; |
|  | default: |
|  | printf("Invalid Choice\n"); |
|  | } |
|  |  |
|  | } while(1); |
|  | return 0; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | /\* |
|  |  |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 2 |
|  | Enter element to be inserted |
|  | 46 |
|  | Your element is successfully inserted |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 2 |
|  | Enter element to be inserted |
|  | 79 |
|  | Your element is successfully inserted |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 2 |
|  | Enter element to be inserted |
|  | 46 |
|  | Your element is successfully inserted |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 1 |
|  | Element at front of queue : 46 |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 4 |
|  | 46 79 46 |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 3 |
|  | Element deleted is 46 |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 6 |
|  | Invalid Choice |
|  |  |
|  | Display Menu |
|  | 1.Peek |
|  | 2.Insert |
|  | 3.Delete |
|  | 4.Display |
|  | 5.Exit |
|  | Enter choice |
|  | 5 |
|  |  |
|  |  |
|  | ------------------ |
|  | (program exited with code: 0) |
|  | Press return to continue |
|  |  |
|  | \*/ |