TOPIC NO.: 13

TITLE: Queue

Program Code:

#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 100

struct Queue {

int items[MAX\_SIZE];

int front;

int rear;

};

void initializeQueue(struct Queue \*q) {

q->front = -1;

q->rear = -1;

}

int isFull(struct Queue \*q) {

return (q->rear == MAX\_SIZE - 1);

}

int isEmpty(struct Queue \*q) {

return (q->front == -1 && q->rear == -1);

}

void enqueue(struct Queue \*q, int value) {

if (isFull(q)) {

printf("Queue is full. Cannot enqueue.\n");

return;

}

if (isEmpty(q)) {

q->front = 0;

q->rear = 0;

}

else {

q->rear++;

}

q->items[q->rear] = value;

}

int dequeue(struct Queue \*q) {

int removeditem;

if (isEmpty(q)) {

printf("Queue is Empty. Cannot Dequeue");

return -1;

}

removeditem = q->items[q->front];

if (q->front == q->rear) {

q->front = -1;

q->rear = -1;

}

else {

DATE: 24.04.24

q->front++;

}

return removeditem;

}

int front(struct Queue \*q) {

if (isEmpty(q)) {

printf("Queue is Empty");

return -1;

}

return q->items[q->front];

}

int rear(struct Queue \*q) {

if (isEmpty(q)) {

printf("Queue is Empty.");

return -1;

}

return q->items[q->rear];

}

void printQueue(struct Queue \*q) {

int i;

if (isEmpty(q)) {

printf("Queue is empty.");

return;

}

printf("Queue : ");

for (i = q->front; i <= q->rear; i++) {

printf(" %d", q->items[i]);

}

printf("\n");

}

int main() {

struct Queue q;

initializeQueue(&q);

int choice, value;

while (1) {

printf("\n\nQueue Operations:\n");

printf("1. Enqueue\n");

printf("2. Dequeue\n");

printf("3. Front element\n");

printf("4. Rear element\n");

printf("5. Display queue\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter value to enqueue: ");

scanf("%d", &value);

enqueue(&q, value);

break;

case 2:

dequeue(&q);

break;

case 3:

printf("Front element: %d\n", front(&q));

break;

case 4:

printf("Rear element: %d\n", rear(&q));

break;

case 5:

printQueue(&q);

break;

case 6:

printf("Exiting...\n");

exit(0);

break;

default:

printf("Invalid choice! Please enter a number between 1 to 6.\n");

}

}

return 0;

}

Output:

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 1

Enter value to enqueue: 6

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 1

Enter value to enqueue: 4

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 5

Queue : 6 4

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 3

Front element: 6

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 4

Rear element: 4

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 2

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 5

Queue : 4

Queue Operations:

1. Enqueue

2. Dequeue

3. Front element

4. Rear element

5. Display queue

6. Exit

Enter your choice: 6

Exiting...