# **Name: Abdurrahman Qureshi**

# **Roll no: 210451**

**Aim: Queue via array**

**CODE:**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int info;

struct node \*ptr;

}\*front,\*rear,\*temp,\*front1;

//int frontelement();

void insertinQueueNode(int data);

void deletinQueueNode();

void empty();

void displayQueue();

void create();

// void queuesize();

int count = 0;

int main()

{

int no, ch, e;

printf("\n\t\t 1 - Insert a Node");

printf("\n\t\t 2 - Delete a Node");

printf("\n\t\t 3 - Front element");

printf("\n\t\t 4 - Empty");

printf("\n\t\t 5 - Exit");

printf("\n\t\t 6 - Display Queue");

printf("\n\t\t 7 - Queue size");

create();

while (1)

{

printf("\n\t\t Enter choice : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("\t\tEnter data : ");

scanf("%d", &no);

insertinQueueNode(no);

break;

case 2:

deletinQueueNode();

break;

case 3:

e = frontelement();

if (e != 0)

printf("\t\tFront element : %d", e);

else

printf("\n\t\t No front element in Queue as queue is empty");

break;

case 4:

empty();

break;

case 5:

exit(0);

case 6:

displayQueue();

break;

case 7:

queuesize();

break;

default:

printf("\t\tWrong choice, Please enter correct choice ");

break;

}

}

return 0;

}

/\* Create an empty queue \*/

void create()

{

front = rear = NULL;

}

/\* Returns queue size \*/

void queuesize()

{

printf("\n\t\t Queue size : %d", count);

}

/\* insertinQueueNodeueing the queue \*/

void insertinQueueNode(int data)

{

if (rear == NULL)

{

rear = (struct node \*)malloc(1\*sizeof(struct node));

rear->ptr = NULL;

rear->info = data;

front = rear;

}

else

{

temp=(struct node \*)malloc(1\*sizeof(struct node));

rear->ptr = temp;

temp->info = data;

temp->ptr = NULL;

rear = temp;

}

count++;

}

/\* displayQueueing the queue elements \*/

void displayQueue()

{

front1 = front;

if ((front1 == NULL) && (rear == NULL))

{

printf("\t\tQueue is empty");

return;

}

while (front1 != rear)

{

printf("\t\t %d ", front1->info);

front1 = front1->ptr;

}

if (front1 == rear)

printf("%d", front1->info);

}

/\* deletinQueueNodeueing the queue \*/

void deletinQueueNode()

{

front1 = front;

if (front1 == NULL)

{

printf("\n\t\t Error: Trying to displayQueue elements from empty queue");

return;

}

else

if (front1->ptr != NULL)

{

front1 = front1->ptr;

printf("\n\t\t deletinQueueNodeued value : %d", front->info);

free(front);

front = front1;

}

else

{

printf("\n\t\t deletinQueueNodeued value : %d", front->info);

free(front);

front = NULL;

rear = NULL;

}

count--;

}

/\* Returns the front element of queue \*/

int frontelement()

{

if ((front != NULL) && (rear != NULL))

return(front->info);

else

return 0;

}

/\* displayQueue if queue is empty or not \*/

void empty()

{

if ((front == NULL) && (rear == NULL))

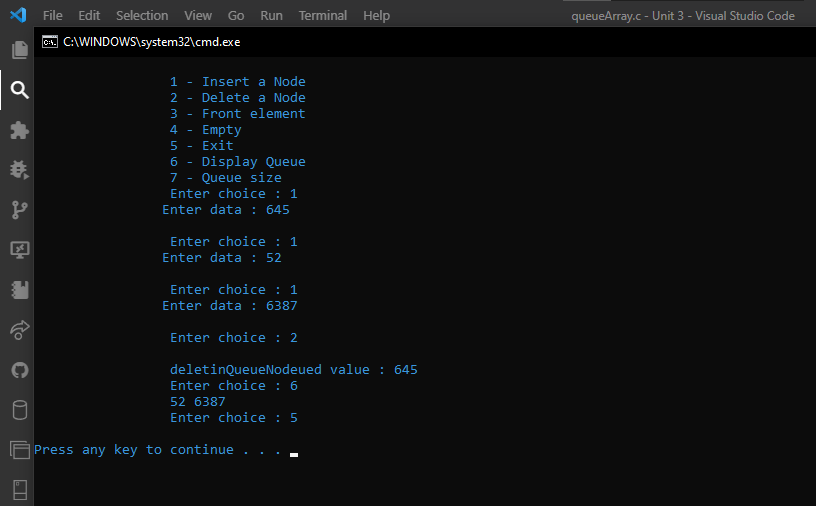
printf("\n\t\t Queue empty");

else

printf("\t\tQueue not empty");

}

**OUTPUT:**



**CODE:**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int info;

struct node \*ptr;

} \* front, \*rear, \*temp, \*front1;

int frontelement();

void enq(int data);

void deq();

void display();

void create();

int count = 0;

int main()

{

int no, ch, e;

printf("1 - Enqueue");

printf(" 2 - Dequeue");

printf("3 - Display");

printf("4 - Exit");

printf("5-front");

create();

while (1){

printf("Enter choice : ");

scanf("%d", &ch);

switch (ch){

case 1:

printf("Enter data : ");

scanf("%d", &no);

enq(no);

break;

case 2:

deq();

break;

case 3:

display();

break;

case 4:

exit(0);

break;

case 5:

e = frontelement();

if (e != 0)

printf("Front element : %d", e);

else

printf("No front element in Queue");

break;

default:

printf("Wrong choice, Try again ");

break;

}

}

return 0;

}

void enq(int data)

{

if (rear == NULL)

{

rear = (struct node \*)malloc(1 \* sizeof(struct node));

rear->ptr = NULL;

rear->info = data;

front = rear;

}

else

{

temp = (struct node \*)malloc(1 \* sizeof(struct node));

rear->ptr = temp;

temp->info = data;

temp->ptr = NULL;

rear = temp;

}

count++;

}

void display()

{

front1 = front;

if ((front1 == NULL) && (rear == NULL))

{

printf("Queue is empty");

return;

}

while (front1 != rear)

{

printf("%d ", front1->info);

front1 = front1->ptr;

}

if (front1 == rear)

printf("%d", front1->info);

}

void deq()

{

front1 = front;

if (front1 == NULL)

{

printf("Error");

return;

}

else if (front1->ptr != NULL)

{

front1 = front1->ptr;

printf("Dequeued value : %d", front->info);

free(front);

front = front1;

}

else

{

printf(" Dequeued value : %d", front->info);

free(front);

front = NULL;

rear = NULL;

}

count--;

}

int frontelement()

{

if ((front != NULL) && (rear != NULL))

return (front->info);

else

return 0;

}

**OUTPUT:**

