**ASSIGNMENT – 5**

1. Write a Menu driven C program to accomplish the following functionalities in Queue using an Array:

a. Insert an element into the queue using an array (Enqueue Operation).

b. Delete an element from the queue using an array (Dequeue Operation).

c. Return the value of the FRONT element of the queue (without deleting it from

the queue) using an array (Peep operation).

d. Display the elements of a queue using an array.

#include <stdio.h>

#include <math.h>

#include <string.h>

#include <stdlib.h>

#define PI 3.14159

int main()

{

int n;

printf("Enter Number of Elements: ");

scanf("%d", &n);

int ar[n];

int ch;

int front = -1, rear = -1;

while (1)

{

printf("1. Enqueue\n2. Dequeue\n3. Peek:\n");

scanf("%d", &ch);

switch (ch)

{

case 1:

if (rear == n - 1)

{

printf("Queue is Full\n");

}

else if (rear == -1)

{

front = 0;

rear = 0;

printf("Enter Data: ");

scanf("%d", &ar[rear]);

}

else

{

rear++;

printf("Enter Data: ");

scanf("%d", &ar[rear]);

}

break;

case 2:

if (front == -1)

{

break;

}

else if (front == rear)

{

front = -1;

rear = -1;

}

else

{

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

{

ar[i] = ar[i + 1];

}

rear--;

}

break;

case 3:

if (front != -1)

{

printf("The First Value of the Queue is: %d\n", ar[front]);

}

else

{

printf("Queue is Empty\n");

}

break;

}

if (front == -1)

{

printf("Queue is Empty\n");

}

else

{

printf("Your Queue:\n");

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

{

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

}

printf("\n");

}

int x;

printf("Do you want to modify the Queue?\n1. Yes\n2. No:\n");

scanf("%d", &x);

if (x == 2)

{

exit(0);

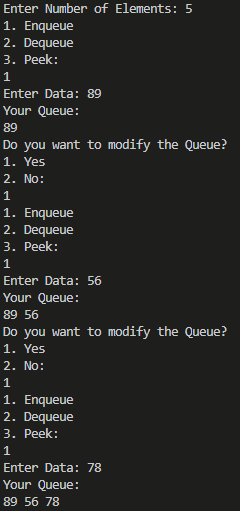
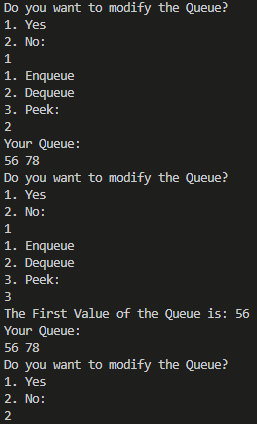
}

}

return 0;

}

Output:

1. Write a Menu driven C program to accomplish the following functionalities in Queue using Linked List:

e. Insert an element into the queue using a Linked List (Enqueue Operation).

f. Delete an element from the queue using a Linked List (Dequeue Operation).

g. Return the value of the FRONT element of the queue (without deleting it from

the queue) using a Linked List (Peep operation).

h. Display the elements of a queue using a Linked List.

#include <stdio.h>

#include <math.h>

#include <string.h>

#include <stdlib.h>

#define PI 3.14159

struct node

{

int data;

struct node \*link;

};

struct node \*enqueue(struct node \*head, int n, int var)

{

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

temp->data = var;

temp->link = NULL;

if (head == NULL)

{

head = temp;

}

else

{

struct node \*loop = head;

int count = 0;

while (loop->link != NULL)

{

count++;

loop = loop->link;

}

count++;

if (count >= n)

{

printf("Queue is Full\n");

}

else

{

loop->link = temp;

}

}

return head;

}

struct node \*dequeue(struct node \*head)

{

if (head == NULL)

{

printf("Queue is Empty\n");

}

else if (head->link == NULL)

{

head = NULL;

}

else

{

head = head->link;

}

return head;

}

void peek(struct node \*head)

{

if (head == NULL)

{

printf("Queue is Empty\n");

}

else

{

printf("The First Value of the Queue is: %d\n", head->data);

}

}

void display(struct node \*head)

{

if (head == NULL)

{

printf("Queue is Empty\n");

}

else

{

printf("Your Queue is:\n");

struct node \*loop = head;

while (loop != NULL)

{

printf("%d ", loop->data);

loop = loop->link;

}

printf("\n");

}

}

int main()

{

int n;

printf("Enter Number of Elements: ");

scanf("%d", &n);

struct node \*head = NULL;

int ch, num;

while (1)

{

printf("1. Enqueue\n2. Dequeue\n3. Peek:\n");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("Enter Data: ");

scanf("%d", &num);

head = enqueue(head, n, num);

break;

case 2:

head = dequeue(head);

break;

case 3:

peek(head);

break;

}

display(head);

int x;

printf("Do you want to modify the Queue?\n1. Yes\n2. No:\n");

scanf("%d", &x);

if (x == 2)

{

exit(0);

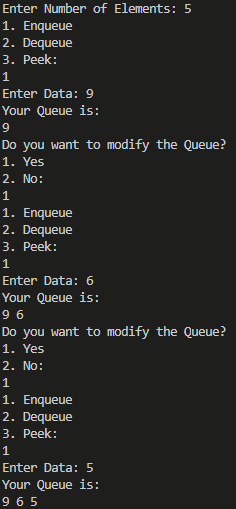
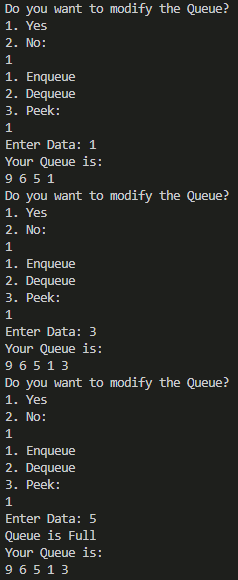
}

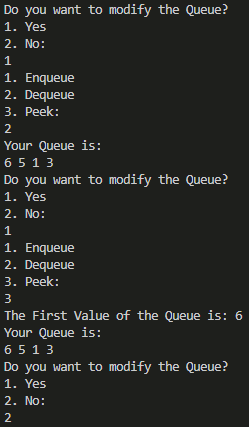
}

return 0;

}

Output:



1. Write a Menu driven C program to accomplish the following functionalities in Circular Queue using Array:

i. Insert an element into the circular queue.

j. Delete an element from the circular queue.

k. Return the value of the FRONT element of the circular queue (without deleting it

from the queue).

l. Display the elements of a circular queue using the circular queue.

#include <stdio.h>

#include <math.h>

#include <string.h>

#include <stdlib.h>

#define PI 3.14159

struct node

{

int data;

struct node \*link;

};

struct node \*enqueue(struct node \*head, int n, int var)

{

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

temp->data = var;

temp->link = temp;

if (head == NULL)

{

head = temp;

}

else

{

struct node \*loop = head;

struct node \*flag = head;

int count = 0;

while (loop->link != flag)

{

loop = loop->link;

count++;

}

count++;

if (count >= n)

{

printf("Queue is Full\n");

}

else

{

loop->link = temp;

temp->link = flag;

}

}

return head;

}

struct node \*dequeue(struct node \*head)

{

if (head == NULL)

{

printf("Queue is Empty\n");

}

else if (head->link == head)

{

head = NULL;

}

else

{

struct node \*loop = head;

while (loop->link != head)

{

loop = loop->link;

}

head = head->link;

loop->link = head;

}

return head;

}

void peek(struct node \*head)

{

if (head == NULL)

{

printf("Queue is Empty\n");

}

else

{

printf("The First Element of the Queue is: %d\n", head->data);

}

}

void display(struct node \*head)

{

if (head == NULL)

{

printf("Queue is Empty\n");

}

else

{

printf("Your Queue is: \n");

struct node \*loop = head;

do

{

printf("%d ", loop->data);

loop = loop->link;

} while (loop != head);

printf("\n");

}

}

int main()

{

int n;

printf("Enter Number of Elements: ");

scanf("%d", &n);

struct node \*head = NULL;

int ch, num;

while (1)

{

printf("1. Enqueue\n2. Dequeue\n3. Peek:\n");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("Enter Data: ");

scanf("%d", &num);

head = enqueue(head, n, num);

break;

case 2:

head = dequeue(head);

break;

case 3:

peek(head);

break;

}

display(head);

int x;

printf("Do you want to modify the Queue?\n1. Yes\n2. No:\n");

scanf("%d", &x);

if (x == 2)

{

exit(0);

}

}

return 0;

}

Output:

