

Lab - 4 }

Bafna Gold
Date: / / Page: / 24

// circular queue

```
#include <stdio.h>
```

```
#define size 5
```

```
void insertq(int[], int);
```

```
void deleteq(int[]);
```

```
void display(int[]);
```

```
int front = -1;
```

```
int rear = -1;
```

```
int main()
```

```
{
```

```
int n, ch;
```

```
int queue[size];
```

```
do
```

```
{
```

```
printf("In In circular queue menu : In
```

```
1. Insert In 2. Delete In 3. Display
```

```
4. Exit (n);
```

```
printf("In Enter your choice : ");
```

```
scanf("%d", &ch);
```

```
switch(ch)
```

```
{
```

case 1:

```
printf("In Enter the inserting  
number : ");
```

```
scanf("%d", &n);
```

```
break;
```

case 2 :

 deleteq(queue);
 break;

case 3 :

 display(queue);
 break;

case 4 :

 exit(0);

default :

 printf("wrong choice: please
 see the options");

{ while(ch != 0);

void insertq(int queue[], int item)

{ if ((front == 0) && rear == size - 1) {

 printf("queue is full, overflow! ");
 return;

{

 printf("queue is full, overflow! ");
 return;

{

 else if (rear == n - 1)

 {

 rear++;

 front++;

{

```
else if (rear == size - 1 && front > 0)
```

```
{
```

```
    rear = 0;
```

```
}
```

```
else
```

```
    rear++;
```

```
{
```

```
    queue[rear] = item;
```

```
void display(int queue[])
```

```
{
```

```
int i;
```

```
printf("\n");
```

```
if (front > rear) return;
```

```
{
```

```
for (i = front; i < size; i++)
```

```
{
```

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

```
}
```

```
for (i = 0; i <= rear; i++)
```

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

```
{
```

```
else
```

```
{
```

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

```
{
```

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

```
}
```

```
{
```

```
{
```

```

Void deleteq (int queue[3])
{
    if (front == -1)
        printf ("queue is empty, Underflow!");
    else if (front == rear)
        printf ("node deleted", queue[front]);
    front = -1;
    rear = -1;
}
else
    printf ("node deleted", queue[front]);
    front++;
}

```

Output :-

Circular Queue menu:

1. Insert

2. Delete

3. Display

4. Exit

Enter Your choice : 2

Queue is empty, Under Flow !!

Enter Your choice : 1

Enter the Inserting number : 1

Enter the choice : 1

Enter the Inserting number : 2

Enter the choice : 1

Enter the Inserting number : 3

Enter the choice : 1

Enter the Inserting number : 4

Enter the choice : 1

Enter the Inserting number : 5

Enter the choice : 1

Enter the Inserting number : 6

queue is full, overflow !

Enter the choice : 3

1 2 3 4 5

← SPONASH

← SPONASH

Enter the choice : 6

wrong choice: please see the

SPONASH options

(Sponash Bar chart) available now

; l " + t e s t b a r s n p } f m n }

(sum = 6 bars) sum

6 bars ← sponash bar } sum

; from ← sponash bar

// Linked List

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Node
```

```
{
```

```
    int data;
```

```
    struct Node* next;
```

```
};
```

```
void insert(struct Node **head, int data)
```

```
{
```

```
    struct Node *newnode = (struct Node*) malloc(sizeof(struct Node));
```

```
    newnode->
```

```
    data = data;
```

```
    newnode->
```

```
    next = *head;
```

```
*head = newnode;
```

```
void display(struct Node *node)
```

```
{
```

```
    printf("In Linked List: ");
```

```
    while (node != NULL)
```

```
{
```

```
    printf("%d", node->data);
```

```
    node = node->next;
```

```
}
```

```
printf("\n");
```

?

```
void main( )
```

?

```
struct Node * head = NULL;
```

```
insert(&head, 4);
```

```
insert(&head, 3);
```

```
insert(&head, 2);
```

```
insert(&head, 1);
```

```
display(head);
```

?

Output :-

~~Linked List :- 1 2 3 4~~

~~Q.1
Ans:-~~