NAME: G.Nithish

REG No: 19BCS0012

Course: DATA structures.

Digital Assignment -> 3

1) Write a menu driven program to implement a queue data structure using Linked List Considering following options in the menu.

a). Create an empty queue

b). Returns queue size

c). Enqueue an elements in the queue

d) Delet an element in the queue

e) Displaying the queue elements.

f) Returns the front element of queue.

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
~~#define capacity 5~~
int count=0;
void create();
void append();
void delete queue();
void display(int n);
```

```
struct node
{ int data;
struct node*next;
}
struct node*head;
struct node*newnode;
struct node *temp;
int main()
{ int x;
```

```
Printf("Queue Program
        using linklisted");
int g=1;
do
{
Printf("\n1> ~~Insert~~ Creat an Empty
        Queue");
Printf("\n 2>DELETE first
        elements in queue");
Printf("\n3> ENQUE an
        Element in the queue")
Printf("\n4> Return queue
        Size");
Printf("\n 5> Display
        the queue elements")

Printf("\n 6> Returns the front
        elements of queue");
int choice;
Scanf("%d",&choice);
Switch (Choice)
{
  case 1:
    if(g==1) {
```

```
  Create (g);
  g++; }
else
   append();
Printf("DONE");
 break;

Case 2:
if (count==0)
  Printf("The Queue is
          EMPTY");
else
   deletequeue();
break;

Case 3;
if ~~(count=~~.
  if (g==1) {
     create(g);
     g++; }
  else
     append();
  Printf("Done");
break;
```

```
case 4:
 Printf("\nThe Queue size is
  %d", count);
  break;

case 5:
 if(count==0)
 { Printf("Queue empty");
 else  display Queue(count);
  break;

case 6:
  if(count==0)
  Printf("Queue is
          Empty");
 else
    display(1);
  break;

 default:
 Printf("ENTER Queue\n");
  break;
 } } while(1);
 getch();
```

```
return; }

Void create(int l)
{
 newnode=(struct node*)
   malloc(size of (struct
                    node));

Printf("Enter the data");
Scanf("%d", &newnode->data);
newnode->next=NULL;
head=newnode;
  temp=newnode;
 count++;

}
Void append()
{ Printf("Enter data:");

 int g;
 Scanf("%d", &g);
 newnode=(struct node*)
   malloc(sizeof(struct node))
   temp=head;
```

Scanned by TapScanner

```
while(temp->next!=NULL)
{
 temp=temp->next;
 }
 temp->next=newnode;
 newnod->data=g;
 newnode->next=NULL;
 Count++; }
 void deletequeue()
{
  struct node *p;
  P=head;
  head=head->next;
  free(p);
  Printf("\nDone");
  Count--;
 }
 void display(int n)
 {
   Printf("\n Created Queu
         \n");
   int i;
   temp=head;

if(n==1)
{
 Printf("%d\n",temp->da
 }
else
{
 while(temp!=NULL)

{ Printf("%d\n",temp->d
temp=temp->next;
} } }
```

19BCS0012

Nithish·G

In a vertical parking lot, totally 5 cars are paked one after the other from one end where the other end is a dead end. If the person who has parked first wants to come out, what kind of ADT he has to follow. Write a C code to implement the above scenario.

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
# define capacity 5
int stack[capacity];
int top=-1;
void push(int element);
int pop();
void peek();
void display();
```

```
int main()
{
 int ch ,~~d~~info;
while(1)
{
 Printf("Vertical Car Parking Max 5cars \n");
 Printf("1. PARK Car\n");
 Printf("\n2. Remove Car\n");
 Printf("\n 3. Top most car\n");
 Printf("\n 4. Display CARS\n");
 Printf("\n 5. Size\n");
 Printf("\n 6. Exit\n");
 Printf("Enter choice: ");
  Scanf("%d", &ch);
  ~~Switch(choice)~~
  Switch(ch)
  {
  case 1:
   Printf("Enter the car Number to park :");
   Scanf("%d", ~~&d~~ & info);
      ~~Push(d~~ Push(info);
              break;
case 2:
 info = pop();
 Printf("Car Number => %d\n", info);
 break;
case 3:
   Peek();
     break;
 case 4:
     display()
     break;
case 5:
  Printf("No. of cars parked: %d\n", top+1);
  break;
case 6:
  Printf("Exiting.\n");
   exit(0);
   break;
  default:
      Printf("Invalid choice, please try again.\n");
 }  Printf("\n\n");
    return0;
    }
```

```
void Push(int element)
{ ~~if(top>=size~~
if(top>=capacity-1)
{ Printf("Parking slot Full...\n");
}
else {
        top++;
   stop[top]= element;
 Printf("Car Parked...\n");}
}

int POP()
{ if(top<0)
{ Printf("No cars Parked,\n");
    return 0;
 }
 return stack[top--]; }

Void Peek()
{ if(top<0)
  {
    Printf("\nNo class Parked.")
  }
else { Printf("\n The top car is %d", stack
                                      [top
```

```
void display()
{ int i
  if(top<0)
  { printf("No cars Parked");
  }
  else
  { Printf("\nThe cars Parked from top to Bottom:
  for(i=top; i>=0; i--)
  {
    Printf("_\n%d\n_\n", stack[i]);
    }
   }
}.
```

Scanned by TapScanner