Assignment on

Course Tittle: Data Structure Sessional

Course Code: CSE-1316

Submitted to

Debojyoti Biswas

Lecturer

Department of Computer Science and Engineering

Leading University, Sylhet

Submitted by

Name: Khadiza Akther

Student's ID: 2012020295

Section: 3F

Department of Computer Science and Engineering

Date of Submisson: FEB 19, 2021

Scanned with CamScanner

Tasks - 1:

Title:

Write a program that uses function to perform the following operation on singly linked list
(i) creation (ii) Insertion (iii) Deletion (iv) Traversal

Problem Analysis:

This program will be perform the following operation on singly linked list:

(i) creation
(ii) ~~Deletion~~ Insertion
(iii) Deletion
(iv) Traversal

Based on problem, it is required to get the input of value, data, position. The program will need a value other than -1 to create the node. To perform the insertion and deletion, the program will need to take input the position of the singly linked list. In addition, the program will also need the value insert, while traversing, it will display the node data.

So, the input is value, position and data. Node data will be displayed, it should be the output variable.

Scanned with CamScanner

Input variable	processing variable/calculation	output variables	Necessary header files/ functions/ macros
value(int) data(int) position(int)	sizeof, malloc	Node data (int)	stdio.h stdlib.h free() scanf() & printf() for formatted i/o.
			creation(), insertIn() Deletion(), traversal()

Problem :-

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
  {
     int node;
     struct node * next;
  } Node;
Node * head =NULL, * tail =NULL;
int count =0;
Node *creation()
  { int value;
    printf("\n Enter some integer value for create a
                linked list and (-1) for stop linked list
                                                \n");
    scanf("%d",&value);
     head = NULL;
     tail = NULL;
while (value != -1)
  { Node * new_node =(Node*) malloc (sizeof(Node));
    new_node -> data = value;
    new_node -> next = NULL;
    if (head == NULL)
     { head = new_node;
       tail = new_node; }
```

Scanned with CamScanner

```
else {
      tail -> next = new_node;
      tail = new_node;
     } scanf("%d", &value);
   } return head;
   }
int totalNode (Node *q)
 { while (q != NULL)
   {  count++;
      q = q -> next;
    } return count;
   }
Node * Insertion()
{ Node *avail;
  int i = 1, position = 1;
  printf("\n Enter the position where we want to insert
                                    Node = ");
  scanf("%d", & position);
if (position < 0 || position > count)
       { printf("\n Invalid position");
         }
else if (position == 1)
   {   Node * new_node = (Node*) malloc(sizeof(Node));
   printf("\n Enter New node value = ");
   scanf("%d", & new_node -> data);
    new_node -> next = head;
     head = new_node;
      count++;
     }
```

Scanned with CamScanner

```
else
{ Node * new_node =(Node*) malloc(sizeof(Node));
    avail = head;
    while (i< position -1)
     { avail = avail ->next
        i++; }
   printf("\n Enter New Node value = ");
   scanf("%d", & new_node ->data);
    new_node -> next = avail -> next;
    avail -> next = new_node;
    count++; }
    return head; }
Node * Deletion()
{ Node * avail ,* next_node;
  int i =1 , position =0;
  printf ("\n Enter the position where we want to
  remove node = ");
  scanf("%d", & position);
    avail = head;
  if(position <0 || position > count)
     { printf ("\n Invalid position \n"); }
else { while(i< position -1)
  { avail = avail -> next;
    i++; }
    next_node = avail -> next
     avail -> next = next_node -> next;
    free (next_node); } return head; }
```

Scanned with CamScanner

```
void p(Node *q)
  { while(q != NULL)
    { printf("%d", q->data);
        q = q -> next; } }
int main()
 { Node* i = creation();
     totalNode(i);
 if(i==NULL)
   { printf("\n List is NULL \n"); }

else { i = Insertion();
       p(i);
       i = Deletion();
       p(i); }
return 0; }
```

Scanned with CamScanner

Tasks-2 :

Title :

Write a program that uses function to perform the following operation on doubly linked list
(i) creation (ii) Insertion (iii) Deletion (iv) Traversal

Problem Analysis

This program will be perform the following operation on doubly linked list:
(i) creation (ii) Insertion (iii) Deletion (iv) Traversal
Based on problem, it is required to get the input of value, data, position. The program will need a value other than -1 to create the node. To perform the insertion and deletion, the program will need to take input the position of the doubly linked list. In addition, the program will also need the value insert, while traversing, it will display thenode data..
so, the input is value, position and data, Node data will be displayed, it should be the output variable.

Input variable	processing variable/function	output variables	Neeessary header files/ functions/ macros
value (int) Data(int) position (int)	sizeof, malloc	Node data (int)	stdio.h stdlib.h free() scanf() & printf() for formatted i/o. Creation() Insertion() peletion() Traversal() totalNode()

Scanned with CamScanner

problem:

```
#include <stdio.h>
# include< stdlib.h>
typedef struct node
  { int data;
      struct node * prev ;
      struct node * next; } Node;
Node *head =NULL,*tail =NULL;
Node *creation()
  { int value;
printf("\nEnter some integer value for create a linked list
                and(-1) for stop linked list.\n");
        scanf ("%d", &value);
         while(value !=-1)
         { Node *newNode =(Node*) malloc(sizeof(Node));
           newNode ->data =value;
          newNode ->prev =NULL;
          new.Node ->next = NULL;
if (head == NULL)
              { head = tail=newNode; }
else
   { tail ->next =newNode;
     newNode ->prev = tail;
       tail =b newNode; }
   scanf("%d",&value); }
   return head; }
int countNode =0;
 int count = 0;
Node *totalNode (Node *a)
 { a = head;
   while (a != NULL) { count++; a= a->next; }
```

Scanned with CamScanner

```
    printf("\n total node in the list = %d ", count); }
Node *Insertion()
  { int position;
    printf("\n Enter the position where we want to
                    insert new node = ");
     scanf("%d", &position);
     Node *new_node = (Node*) malloc(sizeof(*Node));
     printf("\n Enter node value = ");
     scanf("%d", &new_node -> data);
      new_node -> prev = NULL;
      new_node -> next = NULL;
      Node * temp;
       temp = head;
      if(position < 0 || position > count)
           { printf("\n Invalid position"); }
     else if (position == 1)
          { if(head == NULL)
              { head = tail = new_node;
                 count++; }
            else { head -> prev = new_node;
                   new_node -> next = head;
                   head = new_node;
                   count++; } }
     else { int i = 1;
            while (i < position-1)
              { temp = temp -> next;
                i++; }
            new_node -> prev = temp;
            new_node -> next = temp -> next;
```

```
temp -> next = new_node;
 new_node -> next->prev = new_node; }
   return head; }
Node *deletion()
    { int position, i=1;
      Node *temp;
        printf("\nEnter the position where we want
                          to delete = ");
scanf("%d", &position)
    temp = head;
    while( i < position)
            { temp = temp -> next;
              i++; }
    temp -> prev -> next = temp -> next
     temp-> next -> prev = temp -> prev;
                free (temp); }
void p(Node* a)
    { a = head;
      while(a != NULL)
            { printf("%d", a -> data);
                a = a -> next; }
                }
int main()
   { Node *h = creation();
        h = totalNode(h);
if(h == NULL)
        { printf("\nThe list is empty"); }

else { h = Insertion();
     p(h);
     h = Deletion();
       p(h); }
     return 0; }
```

Scanned with CamScanner

Tasks-3:

Title:

Write a program that uses function to perform the following operation on circular linked list.
(i) Creation (ii) Insertion (iii) Deletion (iv) Traversal

Problem Analysis:

This program will be perform the following operation on circular linked list.
(i) Creation (ii) Insertion (iii) Deletion (iv) Traversal

Based on program, it is required to get the input of value, data and position. The program will need a value other than -1 to create the node. to perform the insertion and deletion, the program will need to take input the position of the circular linked list. In addition will also need the value insert, while traversing, it will display the node data.

Input variable	Processing variables/calculation	output variables.	Neeessary header file/ functions/macros
value(int) Data(int) position(int)	size of, malloc	node data (int)	stdio.h stdlib.h createList(), total Node(), Insert Given Position() Deletion(), Traversal()

Scanned with CamScanner

Problem:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
  { int data;
    struct node *next;
  } Node;
Node *head =NULL, *tail=NULL;
Node *createList()
  { int value;
    printf("\n Enter some integer and (-1) for stop
                                    linked list\n");
    scanf("%d", &value);
    while(head==NULL)
        { head=tail=new_node; }
  else while(value!=-1)
    { Node *new_node = (Node*) malloc
                        (sizeof(Node));
      new_node->data = value;
      new_node->next = NULL;
      if(head==NULL)
          { head=tail=new_node; }
      else
          { tail->next=new_node;
            tail=new_node; }
      tail->next = head;
      scanf("%d", &value); }
      return head; }
```

Scanned with CamScanner

```
int count =0;
Node *totalNode()
{ Node *avail;
    avail = head;
    while (avail -> next != head)
    { count++;
      avail = avail -> next; }
    count++;
    printf("\n total Node = %d", count); }
Node *InsertGiven Position()
{ Node *new_node = (Node*) malloc (sizeof(Node));
  int position =0, i=1;
  printf("\n Enter position where we want to
                insert new node = ");
  scanf("%d", &position);
if(position <0 || position > count)
  { printf("\n Invalid position"); }
else if (position == 1)
  { printf("\n Enter the value of Node = ");
    scanf("%d", & new_node -> data);
    new_node -> next = NULL;
if( tail == NULL)
  { tail = new_node;
    tail -> next = new_node; }

else
  { new_node -> next = head;
    tail -> next = new_node;
    head = new_node; }
  count++;
  }
```

Scanned with CamScanner

```
else
 { printf("\n Enter the value of Node = ");
  scanf("%d", & new_node -> data);
   new_node -> next = NULL;
    Node  *temp = tail -> next;
    While (i < position -1)
      { temp = temp -> next;
        i++; }
    new_node -> next = temp -> next;
     temp -> next = new_node; }
     return head; }
Node * Deleteposition ()
 { Node * nextNode, *currentNode;
   int position = 0; i=1;
   currentNode = tail -> next;
   printf("\n Enter position = ");
   scanf("%d", & position);
 if (position < 0 || positin > count)
  { printf("\n Invalid position"); }
  else if (position == 1)
    { Node * temp;
      temp = head;
 if (tail == 0)
  { printf("\n\t Underflow"); }
else if (temp -> next = temp)
 { tail = 0
    free(temp); }
 else
  { temp = head;
     head = head -> next;
```

Scanned with CamScanner

```
        tail -> next = head;
            free(temp); }
            }

else.
    { while(i<position-1)
        { currentNode = currenNode -> next;
          itt; }
        nextNode = currentNode -> next;
        currentNode -> next = nextNode -> next;
         free(nextNode); }
         return head; }
void Traverse (Node *avail)
    {   if(head==0)
          { printf("\n the data is off "); }
        else
        {   avail = head;
            while(avail -> next != head)
             { printf("%d", avail -> data);
                avail = avail -> next; }
              printf("%d", avail -> data); }
int main()
 { Node *y = createlist();
        y = totalNode();
        y = InsertGivenPosition();
  Traverse(y);
y = DeletePosition();
 Traverse(y);
 return 0; }
```

Scanned with CamScanner

Tasks-4:

Title: Write a program that implement stack using Array.

Problem Analysis:

This program will be perform the following operation
(i) Insertion
(ii) Deletion
(iii) Traversal
Based on the problem, it is required to get input of value and choice. The program should display the stack top. The input variable should be value and choice.

Input variable	Processing variable / calculation	output variables	Necessary headerfiles/ functions/ macros
value(int) choice(int)	Top(int)	Top(int)	~~stdli~~ stdio.h stdlib.h push(), pop() display 0, scanf & printf formatted i/o.

Scanned with CamScanner

problem :-

```
#include<stdio.h>
#include<stdlib.h>
#define N 10
int stack[N];
int top=-1;
void push()
{
    int value;
    printf("\nIt Enter value = ");
    scanf("%d",&value);
  if(top==N-1)
    { printf("\nIt Overflow");
      }
    else
      { top++;
        stack[top]=value;
        }
      }
void pop()
  { int item;
  if(top == -1)
  { printf("\nIt Underflow");
    }
  else
    { item = stack[top];
      top--;
      printf("\n\t popped item =%d",item);
        }}
```

Scanned with CamScanner

```
void display()
    { int i;
      printf("\n Enter stack elements = ");
        for(i=top ; i>0 ; i--)
              { printf("%d", *stack[i]);}
              }
int main()
    { int choice;
      printf("\n 1- Insert stack");
      printf("\n 2- Delete stack");
      printf("\n 3- Display ");
      printf("\n 4- Exit\n");
     while(1)
  { printf("\n Enter your choice = ");
    scanf("%d" , &choice);
    switch(choice)
      { case 1:
                push();
                break;
        case 2:
                pop();
                break;
        case 3:
                display();
                break;
        case 4:
              printf("The program is over..\n");
              Exit(0);
               break;
```

Scanned with CamScanner

```
default:
    printf("Invalid choice, please tray again.\n"); }
    } return 0; }
```

Scanned with CamScanner

Tasks-4.(ii):

Title:- Write a program that implement stack (its operation using Linked List (pointers)

Problem Analysis:-

This program will be perform the following operations. (i) Insertion.
(ii) Deletion
(iii) Traversal

Base on program, it is required to get input of value and choice. The program display the stack top. The input variable should be value and choice.

Input variables	Processing variable/ calculation	output variables	Necessary header file/ functions/ macros
value(int) choice(int)	TOP(int) sizeof, malloc	TOP(int)	stdio.h stdlib.h push(), pop(), display(), scanf() & printf formatted i/o.

Scanned with CamScanner

problem:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
        { int node;
          struct node *Link;
          } Node;
          Node *top=0;
void posh()
    { int value;
        printf ("\n\f Enter value = ");
        scanf("%d", &value);
        Node *new_node =(Node*) malloc(sizeof(Node)
                                        );
      new_node ->data = value;
          new_node -> Link = top;
           top = new_node; }
void pop()
       { Node *temp;
        temp  =top;
         if (top==0)
          { printf("\n\t Underflow");
              }
         else
          { printf("\n\t  popped data = %d", top->
                                          data);
         top =top -> Link;
           free(temp); } }

void display()
     { Node *a ;
        a=top;
```

Scanned with CamScanner

```
if (top==0)
    { printf("\n\tstack empty"); }
else
    { printf("\n Enter stack elements = ");
    while( a!=0)
        { printf(" %d ", a->data);
        a = a->link; } } }
int main()
  { int choice;
    printf("\n 1 - Insert stack");
    printf("\n 2 - Delete stack");
    printf("\n 3 - Display ");
    printf("\n 4 - Exit \n");
    while(1)
      { printf("\n Enter your choice = ");
        scanf("%d", &choice);
        switch (choice)
        {
      case 1:
          push();
          break;
      case 2:
          pop();
          break;
      case 3:
          display();
          break;
      case 4:
          printf("The program is over:\n");
          Exit(0);
          break;
      default:
          printf("Invalid choice, please try
                                  again.\n"); }
      }return 0;
      }
```

Scanned with CamScanner

Tasks-5:

Title:

Write a program that implement queue (it operation) using Array.

Problem Analysis:

This program will be perform the following operation
(i) Insertion.
(ii) Deletion
(iii) Traversal
Based on the problem, it is required to get input of value and choice. The program should display the queue Front and rare. The input variable should be value and choice.

Input variables	Processing variable/ calculation	Output variables	Necessary header files /functions/ macros
value(int) choice(int)	Front (int) rare (int)	Front(int) rare(int)	stdio.h stdlib.h push(), pop(), display(), scanf() & printf() formatted i/o

Scanned with CamScanner

Problem:-

```
#include <stdio.h>
# include<stdlib.h>
 # define N 10
int queue[N];
int front =-1;
int rare =-1;
void push()
{ int value;
    printf("\n\t Enter value= ");
    scanf("%d", &value);
    if (rare == N-1)
            { printf("\nOverflow"); }
    else if (front==-1 && rare ==-1)
        { front = rare=0;
          queue [rare] =value; }
    else
        { rare ++;
          queue [rare] =value; } }
void pop()
{   int item;
    if (front ==-1 && rare = = -1)
        { printf("\n\t underflow"); }
else if (front == rare)
    { front = rare = -1; }
else { printf("\n\t popped value from queue=%d",
                                queue [front]);
       front++; } }
```

Scanned with CamScanner

```
void display()
  { int i;
    if(front ==-1 && rare == -1)
              { printf("\nt queue is Empty"); }
    else { printf("\n Enter stack Elements = ");
           for(i=front ; i< rare+1; i++)
             { printf("%d", queue[i]); } } }
int main()
  { int choice;
    printf("\n 1 - Insert Queue");
     printf("\n 2- Delete Queue");
     printf("\n 3 - Display");
     printf("\n 4- Exit\n");
    while(1){
     printf("\nEnter your choice = ");
     scanf("%d", &choice);
     switch (choice) {

case 1:
       push();
       break;
case 2: pop();
        break;
case 3:
       display();
        break;
case 4:
       printf ("The program is over.\n");
        Exit (0);
         break;
default:
       printf("Invalid choice, please try again.\n");
                                  } } return 0; }
```

Scanned with CamScanner

Task-5(ii):

Title:

Write a program that implement queue (it operation) using linked list.

Problem Analysis:

This program will be perform the following operations (i) Insertion. (ii) Deletion (iii) Traversal

Based on the problem, it is required to get input of value and choice. The program should display the queue front & rare. The input variable should be value and choice.

Input variables	Processing variables/ calculation	output variables	Necessary head files/ functions/ macros
value(int) choice(int)	size of, malloc front(int) rare(int)	front(int) rare(int)	stdio.h stdlib.h push(), pop() display(), scanf() & printf() formatted i/o.

Problem:

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
 2 int data;
  struct node *link;
  }Node;
Node *front =0, *rare=0;
```

Scanned with CamScanner

```
void push()
 { Node *new_node =(Node*) malloc (sizeof(Node));
  printf("\n\t Enter value = ");
  scanf("%d", &new_node->data);
   new_node->Link = NULL;
  if (front==0 && rare==0)
     { front =rare =new_node; }
   else
   { rare->Link=new_node;
      rare = new_node; }
       }
void pop()
 { Node * temp;
   if(front ==0 && rare ==0)
    { printf("\n\t underflow");
   else
   { printf("%d", front->data);
     front = front->link;
      free(temp); }
       }
void display()
 { Node *temp;
    if(front ==0 && rare==0)
     { printf("\n\t overflow"); }

else
  { temp=front;
     while(temp!=0)
       { printf("%d",temp->data);
          temp = temp->link; }
       }
```

Scanned with CamScanner

```
int main()
{int choice;
    printf("\n\t 1- Insert Queue");
    printf ("\n 2- Delete Queue");
    printf("\n 3- Display ");
    printf("\n 4- Exit\n");
while(1){
  printf("\n Enter choice = ");
  scanf("%d", &choice);
  switch (choice)
  { case 1:
          push();
          break;

     case 2:
          pop();
          break;

case 3:
          display();
          break;

 case 4:
          printf("The program is over:\n");
          Exit(0);
          break;
default:
  printf("Invalid choice, please try
                  again.\n"); }

} return 0; }
```

Scanned with CamScanner

Task - 6(i):

Titler Write a program that implement circular Queue using arrays.

Problem Analysis:

~~Based~~ This program will be perform the following operations (i) Insert (ii) Delete (iii) Traverse. Based on the problem, it is required to get input of ~~value~~ choice and item. The program should display the queue front and rare. The input variable should be item and choice.

Input variables	Processing variables/ calculation	output variables	Necessary header files/ functions / macros
choice(int) Item(int)	front(int) rare (int)	equeue_ arr[](int)	stdio.h stdlib.h display() insert() del(), scanf() & printf formatted i/o.

problem :-

```
#include<stdio.h>
# include <stdlib.h>
 # define MAX 10
int  equeue_arr[MAX];
int  front = -1;
 int  rear = -1;
void display();
void insert(int item);
 int del();
```

Scanned with CamScanner

```
int main()
{  int choice, item;
   while(1)
    {  printf("\n 1. Insert\n");
       printf(" 2. Delete\n");
       printf(" 3. Display");
       printf(" 4. Exit\n");
printf("\n Enter your choice: ");
 scanf("%d", &cholee);
switch (choice)
 {  case 1:
       ptintf ("\n Enter the element for
                    insertion: ");
       scanf("%d", &item);
        insert (item);
       break;
case 2:
    printf("\n Element deleted is: %d", del(x));
    break;
case 3:
    display();
     break;
case 4:
     exit(0);
default:
     printf("\n Wrong choice\n"); }
}
    return 0; }
void insert (int item)
  { if ((front == 0 && rear == MAX-1) || (front ==
                                  rear+1))
```

Scanned with CamScanner

```
{ printf("\n Queue overflow \n");
        return; }
if (front == -1) front = 0;
if (rear == MAX-1) rear = 0;
else
        rear = rear+1;
        equeue_arr[rear] = item ;
    }
int del()
{   int item;
    if(front == -1)
        { printf("\n Queue underflow \n");
            Exit (1); }
    item = equeue_arr[front];
    if (front == rear)
        { front = -1; rear=-1 ; }
    else if (front == MAX-1) front=0;
    else front = front+1;
        return item; }

void display ()
{ int i;
  if ( front == -1)
    { printf("\n Queue is empty \n");
        return; }
    printf("\n Queue element : \n");
     i=front ;
   if (front <= rear)
     { while (i<=rear)
          printf("%d", equeue_arr[i++]);
     }
```

Scanned with CamScanner

```
else
  { while (i<=MAX-1)
      printf("%d ", cqueue_arr[i++]);
       i=0;
       while(i<=rear)
            printf("%d", cqueue_arr[i++]); }
        printf("\n"); }
```

Scanned with CamScanner