

CSE 2003: Lab Assignment #6

Due on Thursday, March 9, 2017

Prof. Shaik Naseera 2:00pm

Jacob John

Contents

Problem 1	3
Problem 2	6

Problem 1

Write a program to implement Linear Queue using Linked Lists.

Listing 1: C program to implement Linear Queue using Linked Lists.

```
/*Program of queue using linked list*/
#include<stdio.h>
#include<stdlib.h>
struct node
5 {
    int info;
    struct node *link;
} *front = NULL, *rear = NULL;
void insert(int item);
10 int del();
int peek();
int isEmpty();
void display();

15 int main()
{
    int choice,item;
    while(1)
    {
20         printf("1.Insert\n");
        printf("2.Delete\n");
        printf("3.Display the element at the front\n");
        printf("4.Display all elements of the queue\n");
        printf("5.Quit\n");
25         printf("Enter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
30             printf("Input the element for adding in queue: ");
            scanf("%d",&item);
            insert(item);
            break;

            case 2:
35             printf("Deleted element is %d\n",del());
            break;

            case 3:
40             printf("Element at the front of queue is %d\n",peek());
            break;

            case 4:
            display();
45             break;

            case 5:
            exit(1);
        }
    }
}
```

```
50         default:
            printf("Wrong choice\n");
            /*End of switch()*/
        /*End of while*/
    }/*End of main()*/

55 void insert(int item)
{
    struct node *tmp;
    tmp = (struct node *)malloc(sizeof(struct node));
60     if(tmp == NULL)
    {
        printf("Memory not available\n");
        return;
    }
65     tmp->info = item;
    tmp->link = NULL;
    if(front==NULL) /*If queue is empty()*/
        front = tmp;
    else
70         rear->link = tmp;
    rear = tmp;
}/*End of insert()*/

int del()
75 {
    struct node *tmp;
    int item;
    if(isEmpty())
    {
80         printf("Queue Underflown\n");
        exit(1);
    }
    tmp = front;
    item = tmp->info;
85     front = front->link;
    free(tmp);
    return item;
}/*End of del()*/

90 int peek()
{
    if(isEmpty())
    {
95         printf("Queue Underflow\n");
        exit(1);
    }
    return front->info;
}/*End of peek()*/

100 int isEmpty()
{

```

```

        if(front==NULL)
            return 1;
        else
105         return 0;
    }/*End of isEmpty()*/

void display()
{
110     struct node *ptr;
    ptr = front;
    if (isEmpty())
    {
        printf("Queue is empty\n");
115         return;
    }
    printf("Queue elements: \n\n");
    while(ptr!=NULL)
    {
120         printf("%d ",ptr->info);
        ptr = ptr->link;
    }
    printf("\n\n");
}/*End of display()*/

```

Output:

```

Input the element for adding in queue: 10
1.Insert
2.Delete
3.Display the element at the front
4.Display all elements of the queue
5.Quit
Enter your choice: 1
Input the element for adding in queue: 20
1.Insert
2.Delete
3.Display the element at the front
4.Display all elements of the queue
5.Quit
Enter your choice: 1
Input the element for adding in queue: 30
1.Insert
2.Delete
3.Display the element at the front
4.Display all elements of the queue
5.Quit
Enter your choice: 2
Deleted element is 10
1.Insert
2.Delete
3.Display the element at the front
4.Display all elements of the queue
5.Quit
Enter your choice: 3
Element at the front of queue is 20
1.Insert
2.Delete
3.Display the element at the front
4.Display all elements of the queue
5.Quit
Enter your choice: 4
Queue elements:
20 30
1.Insert
2.Delete
3.Display the element at the front
4.Display all elements of the queue
5.Quit
Enter your choice: 5
Jacobs-MBP:Downloads jacobjohn$

```

```

1  /*Program of queue using linked list*/
2  #include<stdio.h>
3  #include<stdlib.h>
4  struct node
5  {
6      int info;
7      struct node *link;
8  } *front = NULL, *rear = NULL;
9  void insert(int item);
10 void del();
11 int peek();
12 int isEmpty();
13 void display();
14
15 int main()
16 {
17     int choice,item;
18     while(1)
19     {
20         printf("1.Insert\n");
21         printf("2.Delete\n");
22         printf("3.Display the element at the front\n");
23         printf("4.Display all elements of the queue\n");
24         printf("5.Quit\n");
25         printf("Enter your choice: ");
26         scanf("%d",&choice);
27         switch(choice)
28         {
29             case 1:
30                 printf("Input the element for adding in queue: ");
31                 scanf("%d",&item);
32                 insert(item);
33                 break;
34
35             case 2:
36                 printf("Deleted element is %d\n",del());
37                 break;
38
39             case 3:
40                 printf("Element at the front of queue is %d\n",peek());
41                 break;
42
43             case 4:
44                 display();
45                 break;
46
47             case 5:
48                 exit(1);

```

Line: 19:3 | C | Tab Size: 4 | main

Problem 2

Write a program to implement Stacks using Linked Lists.

Listing 2: C program to implement Stacks using Linked Lists

```
/*Program of stack using linked list*/
#include<stdio.h>
#include<stdlib.h>
struct node
5 {
    int info;
    struct node *link;
} *top = NULL;
void push(int item);
10 int pop();
int peek();
int isEmpty();
void display();

15 int main()
{
    int choice,item;
    while(1)
    {
20         printf("1.Push\n");
        printf("2.Pop\n");
        printf("3.Display item at the top\n");
        printf("4.Display all items of the stack\n");
        printf("5.Quit\n");
25         printf("Enter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
30                 printf("Enter the item to be pushed: ");
                 scanf("%d",&item);
                 push(item);
                 break;

            case 2:
35                 item = pop();
                 printf("Popped item is: %d\n",item);
                 break;

            case 3:
40                 printf("Item at the top is %d\n",peek());

            case 4:
45                 display();
                 break;

            case 5:
                 exit(1);
        }
    }
}
```

```
50         default:
            printf("Wrong choice\n");
            /*End of switch*/
        } /*End of while*/
    } /*End of main() */

55 void push(int item)
{
    struct node *tmp;
    tmp = (struct node *)malloc(sizeof(struct node));
60     if (tmp==NULL)
    {
        printf("Stack Overflow\n");
        return;
    }
65     tmp->info = item;
    tmp->link = top;
    top = tmp;
} /*End of push() */

70 int pop()
{
    struct node *tmp;
    int item;
    if (isEmpty())
75     {
        printf("Stack Underflow\n");
        exit(1);
    }
    tmp = top;
80     item = tmp->info;
    top = top->link;
    free(tmp);
    return item;
} /*End of pop() */

85

int peek()
{
    if (isEmpty())
90     {
        printf("Stack Underflow\n");
        exit(1);
    }
    return top->info;
95 } /*End of peek() */

int isEmpty()
{
    if (top==NULL)
100     return 1;
    else
```

```

        return 0;
    } /*End of isEmpty() */

105 void display()
    {
        struct node *ptr;
        ptr = top;
        if (isEmpty())
110     {
            printf("Stack is empty\n");
            return;
        }
        printf("Stack elements: \n");
115     while (ptr != NULL)
        {
            printf(" %d\n", ptr->info);
            ptr = ptr->link;
        }
120     printf("\n");
    } /*End of display() */

```

Output:

```

3.Display item at the top
4.Display all items of the stack
5.Quit
Enter your choice: 1
Enter the item to be pushed: 20
1.Push
2.Pop
3.Display item at the top
4.Display all items of the stack
5.Quit
Enter your choice: 1
Enter the item to be pushed: 30
1.Push
2.Pop
3.Display item at the top
4.Display all items of the stack
5.Quit
Enter your choice: 2
Popped item is: 30
1.Push
2.Pop
3.Display item at the top
4.Display all items of the stack
5.Quit
Enter your choice: 3
Item at the top is 20
Stack elements:
20
10
1.Push
2.Pop
3.Display item at the top
4.Display all items of the stack
5.Quit
Enter your choice: 4
Stack elements:
20
10
1.Push
2.Pop
3.Display item at the top
4.Display all items of the stack
5.Quit
Enter your choice: 5

```

```

1  /*Program of stack using Linked List*/
2  #include<stdio.h>
3  #include<stdlib.h>
4  struct node
5  {
6      int info;
7      struct node *link;
8  } *top = NULL;
9  void push(int item);
10 int pop();
11 int peek();
12 int isEmpty();
13 void display();
14
15 int main()
16 {
17     int choice,item;
18     while(1)
19     {
20         printf("1.Push\n");
21         printf("2.Pop\n");
22         printf("3.Display item at the top\n");
23         printf("4.Display all items of the stack\n");
24         printf("5.Quit\n");
25         printf("Enter your choice: ");
26         scanf("%d",&choice);
27         switch(choice)
28         {
29             case 1:
30                 printf("Enter the item to be pushed: ");
31                 scanf("%d",&item);
32                 push(item);
33                 break;
34
35             case 2:
36                 item = pop();
37                 printf("Popped item is: %d\n",item);
38                 break;
39
40             case 3:
41                 printf("Item at the top is %d\n",peek());
42
43             case 4:
44                 display();
45                 break;
46
47             case 5:
48                 break;

```