CSE 2003: Lab Assignment #6

Due on Thursday, March 9, 2017

Prof. Shaik Naseera 2:00pm

Jacob John

Contents

Problem 1

Write a program to implement Linear Queue using Linked Lists.

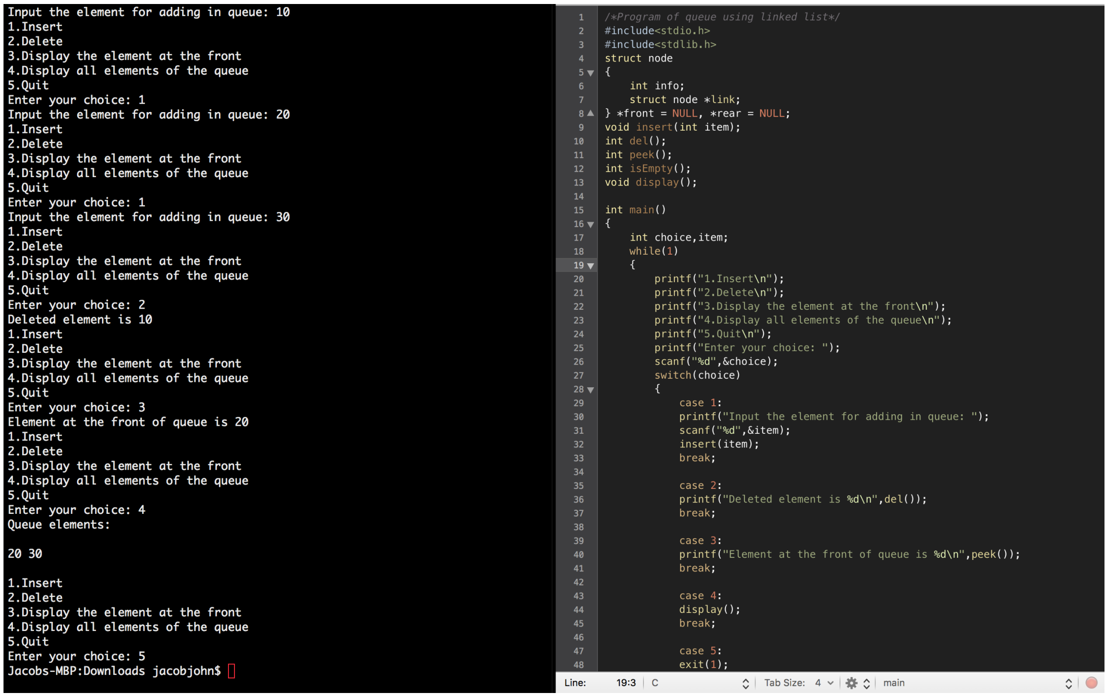
Listing 1: C program to implement Linear Queue using Linked Lists.

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

int main()
{
     int choice,item;
     while(1)
     {
          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");
          printf("Enter your choice: ");
          scanf("%d",&choice);
          switch(choice)
          {
               case 1:
               printf("Input the element for adding in queue: ");
               scanf("%d",&item);
               insert(item);
               break;

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

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

               case 4:
               display();
               break;

               case 5:
               exit(1);
```

Problem 1 continued on next page. . .

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

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

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

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

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

void display()
{
	struct node *ptr;
	ptr = front;
	if (isEmpty())
	{
		printf("Queue is empty\n");
		return;
	}
	printf("Queue elements: \n\n");
	while(ptr!=NULL)
	{
		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$ 
```

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

int main()
{
    int choice,item;
    while(1)
    {
        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");
        printf("Enter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            printf("Input the element for adding in queue: ");
            scanf("%d",&item);
            insert(item);
            break;

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

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

            case 4:
            display();
            break;

            case 5:
            exit(1);
```

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
{
     int info;
     struct node *link;
} *top = NULL;
void push(int item);
int pop();
int peek();
int isEmpty();
void display();

int main()
{
     int choice,item;
     while(1)
     {
          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");
          printf("Enter your choice: ");
          scanf("%d",&choice);
          switch(choice)
          {
               case 1:
               printf("Enter the item to be pushed: ");
               scanf("%d",&item);
               push(item);
               break;

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

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

               case 4:
               display();
               break;

               case 5:
               exit(1);
```

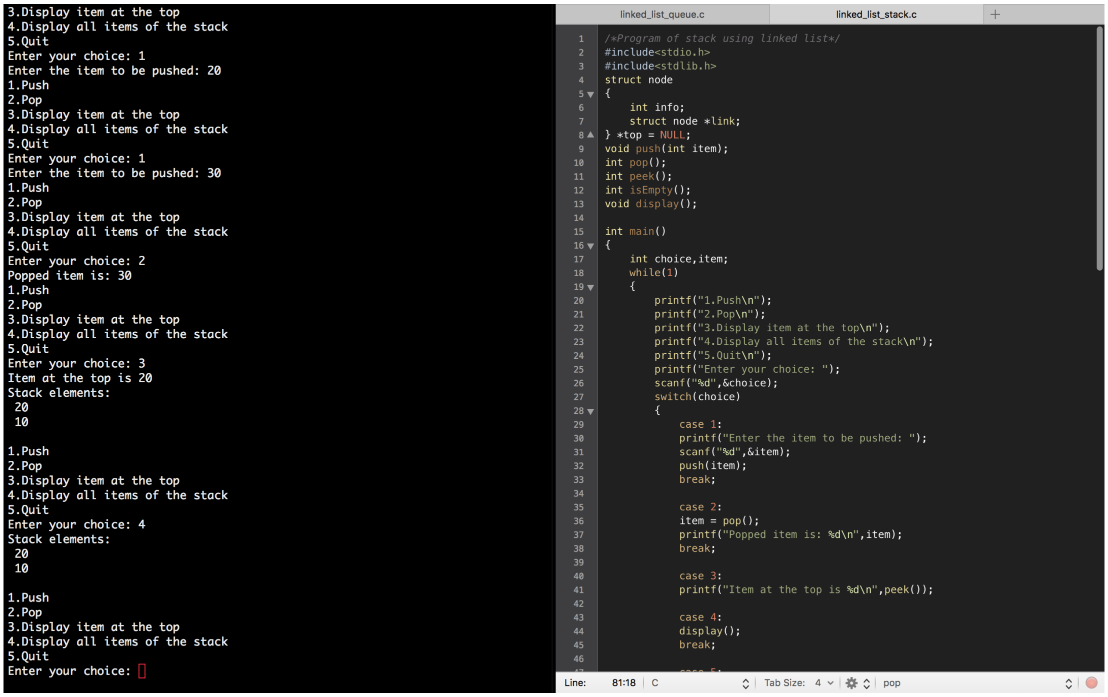
Problem 2 continued on next page. . .

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

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

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


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

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

Problem 2 continued on next page. . .

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

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

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

int main()
{
    int choice,item;
    while(1)
    {
        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");
        printf("Enter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            printf("Enter the item to be pushed: ");
            scanf("%d",&item);
            push(item);
            break;

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

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

            case 4:
            display();
            break;
```