DSAL

Week 7

Udeet Mittal

CSE C3

Roll Number 64

I. SOLVED EXERCISE:

1) Implement stack using singly linked list.

Filename: stack_sll_fun.h

```
#include<stdlib.h>

typedef struct node
{ int info;
struct node *link;
}NODE;
NODE* push(NODE *list,int x)
{ NODE *new,*temp;
new=(NODE*) malloc(sizeof(NODE));
new->link=list;
new->info=x;
return(new);
}
NODE* pop(NODE *list)
{ NODE *prev,*temp;
if(list==NULL)
{
```

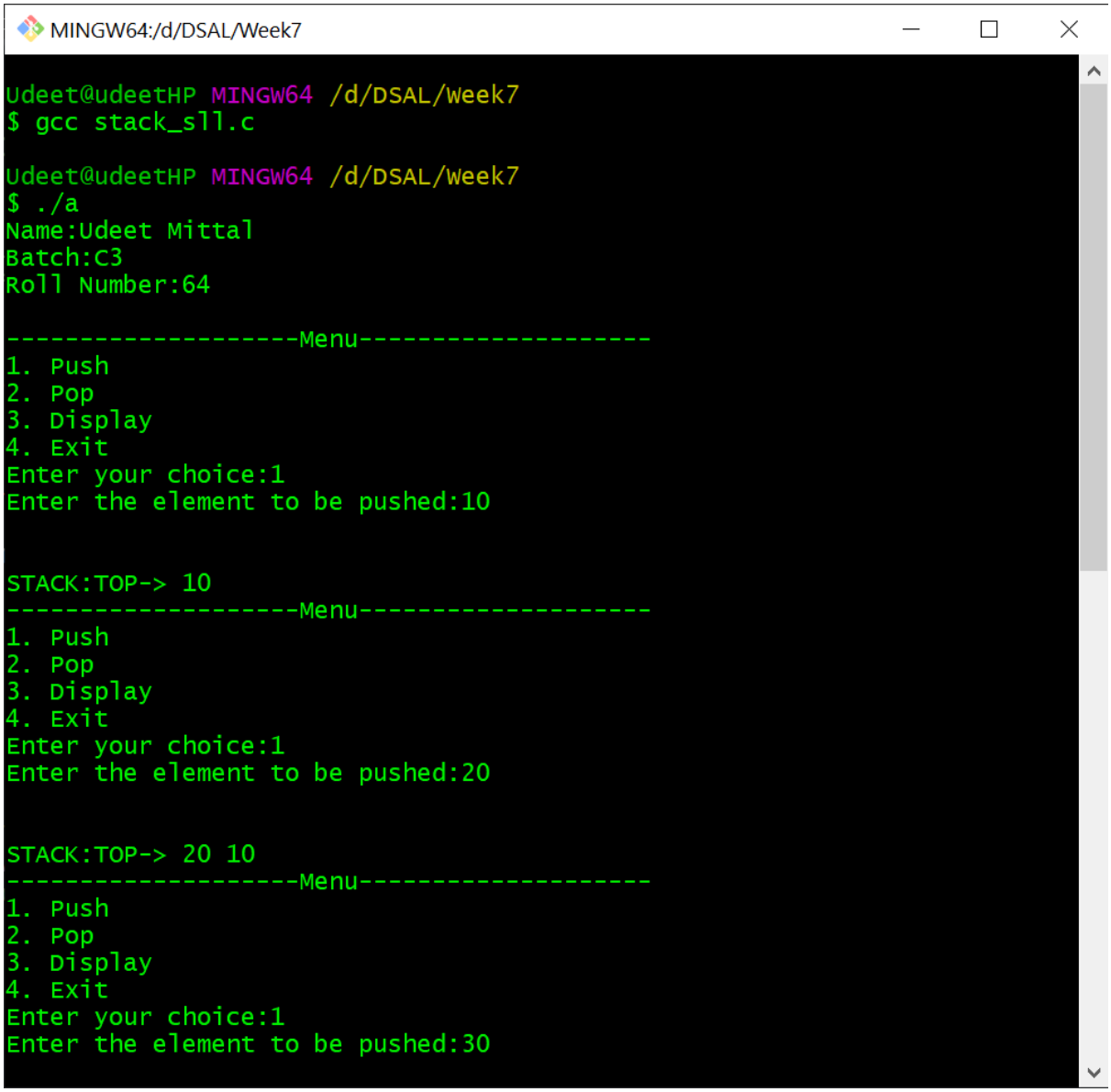
```
printf("\nStack Underflow\n");
return(list);
}
temp=list;
 printf("Deleted element is %d",temp->info);

 list = list->link;
return(list);
}
void display(NODE *list)
{
NODE *temp;
printf("\n\nSTACK:");
if(list==NULL)
{
printf(" Stack is empty");
printf("\n\n**********************************************");
return;
}
printf("TOP-> ");

temp=list;
while(temp!=NULL)
{
printf("%d ",temp->info);
temp=temp->link;
}

}
```

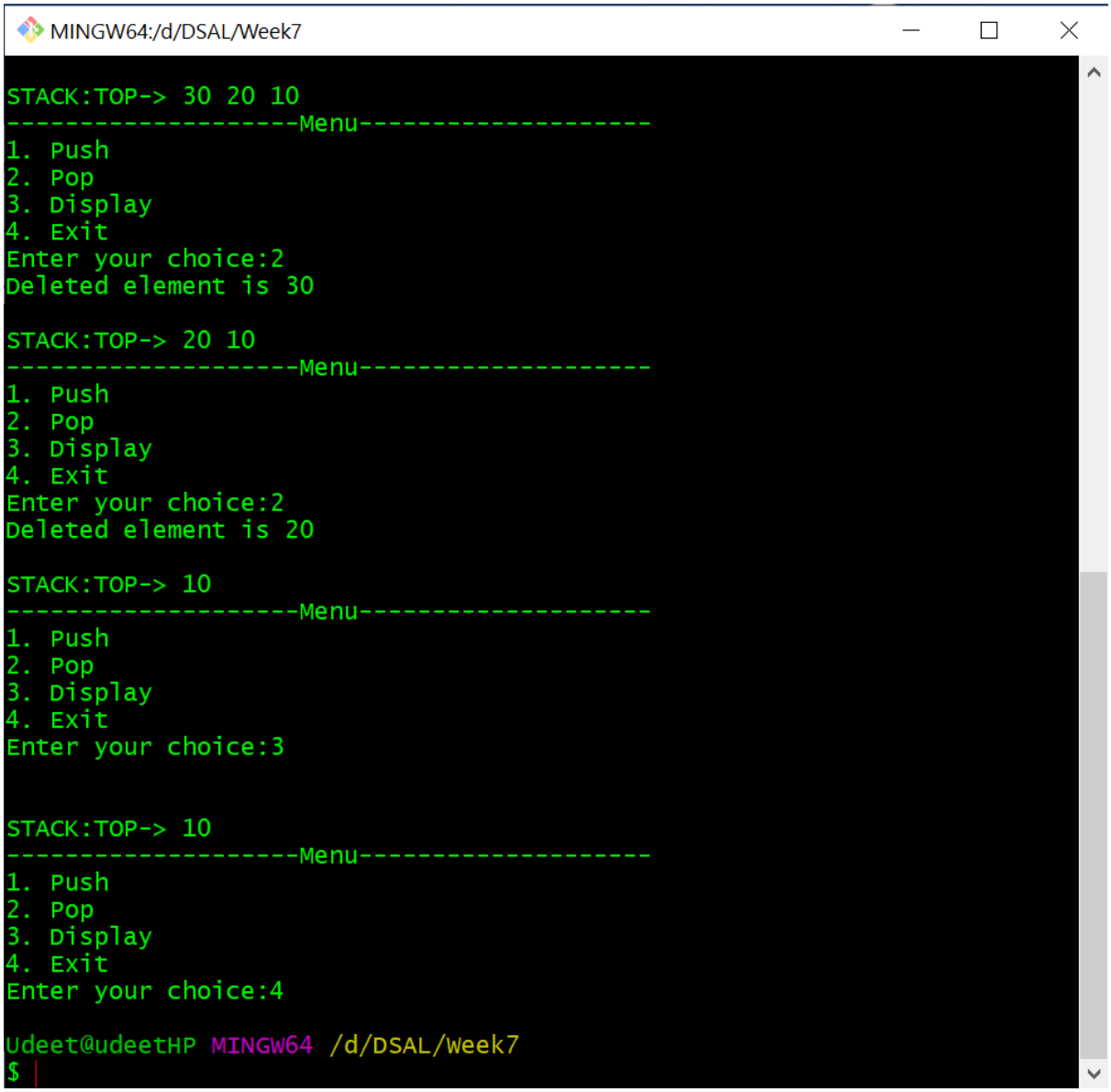
```
int getchoice()
{
int ch;
printf("\n--------------------Menu--------------------\n");
printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");
printf("Enter your choice:");
scanf("%d",&ch);
return(ch);
}
```

Filename: stack_sll.c

```
#include<stdio.h>
#include<stdlib.h>
#include "stack_sll_fun.h"
int main()
{
   printf("Name:Udeet Mittal\nBatch:C3\nRoll Number:64\n");
NODE *list;
int x,ch;
list=NULL;
while(1)
{
ch=getchoice();
switch(ch)
{
case 1: printf("Enter the element to be pushed:");
scanf("%d",&x);
list=push(list,x);
```

```
display(list);
break;
case 2: list=pop(list);
display(list);
 break;
case 3: display(list);
// getc();
break;
case 4: exit(1);
default: printf("\nInvalid choice");
printf("\n\n**********************************************");
}
}
return 0;
}
```


MINGW64:/d/DSAL/Week7
Udeet@udeetHP MINGW64 /d/DSAL/Week7
$ gcc stack_sll.c
Udeet@udeetHP MINGW64 /d/DSAL/Week7
$ ./a
Name:Udeet Mittal
Batch:C3
Roll Number:64
--------------------Menu--------------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice:1
Enter the element to be pushed:10
STACK:TOP-> 10
--------------------Menu--------------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice:1
Enter the element to be pushed:20
STACK:TOP-> 20 10
--------------------Menu--------------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice:1
Enter the element to be pushed:30



2)Given two polynomials, write a program to perform the addition of two polynomials represented using doubly circular linked list with header and display the result.

Filename: poly_add_dll_fun.h

struct node

{ int info;

int ex;

struct node *llink;

struct node *rlink;

```
};
typedef struct node *NODE;
NODE add(NODE head,int n,int e)
{
 NODE temp,last;
temp=(NODE)malloc(sizeof(struct node));
temp->info=n;
temp->ex=e;
last=head->llink;
temp->llink=last;
last->rlink=temp;
temp->rlink=head;
head->llink=temp;
return head;
}
NODE sum(NODE h1,NODE h2,NODE h3)
{
NODE one,two;
one=h1->rlink;
two=h2->rlink;
while(one!=h1 && two!=h2)
{ if((one->ex)==(two->ex))
{ h3=add(h3,((one->info)+(two->info)),one->ex);
one=one->rlink;
two=two->rlink;
}
else if(one->ex>two->ex)
{ h3=add(h3,one->info,one->ex);
one=one->rlink;
```

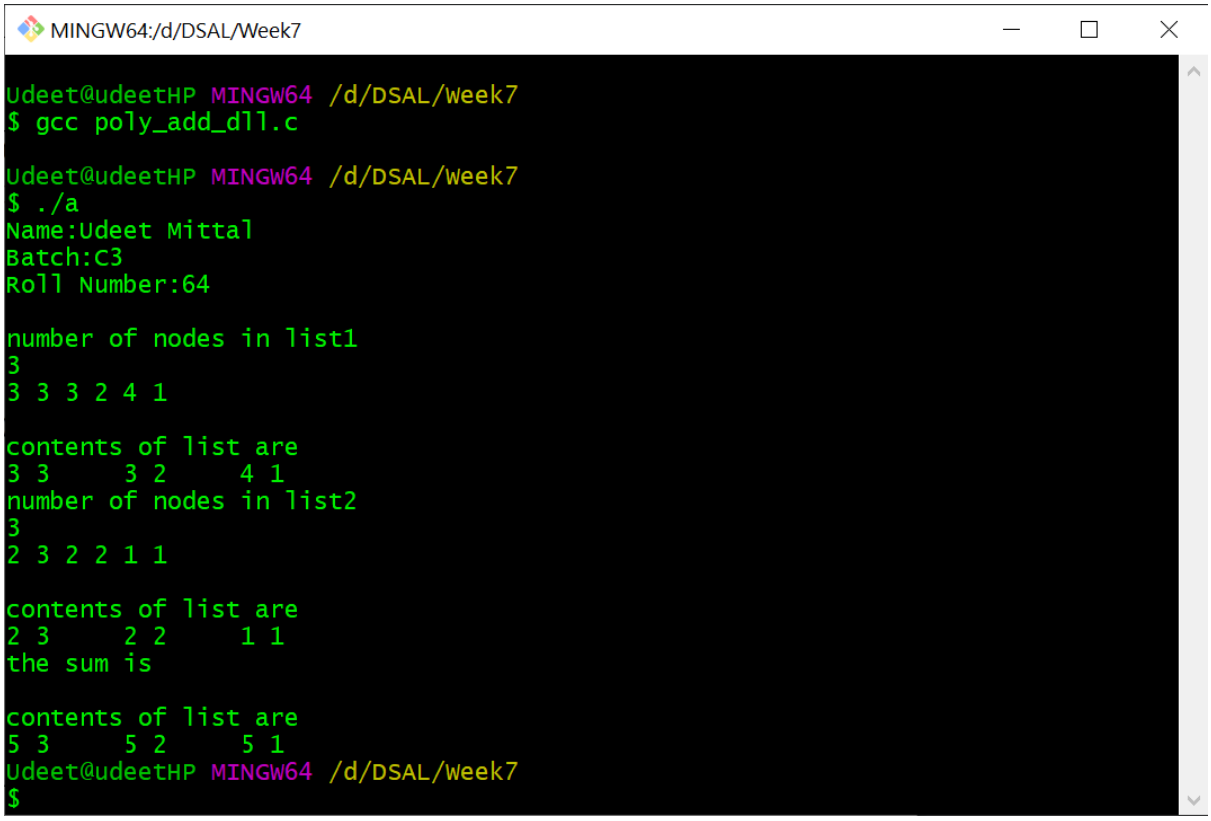
```
}
else
{ h3=add(h3,two->info,two->ex);
two=two->rlink;
}
}
while(two!=h2)
{ h3=add(h3,two->info,two->ex);
two=two->rlink;
}
while(one!=h1)
{ h3=add(h3,one->info,one->ex);
one=one->rlink;
}
return h3;
}
void display(NODE head)
{ printf("\ncontents of list are\n");
NODE temp=NULL;
temp=head->rlink;
while(temp!=head)
{ printf("%d %d\t",temp->info,temp->ex);
temp=temp->rlink;
}}
```

Filename: poly_add_dll.c

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

```
#include "poly_add_dll_fun.h"
int main()
{printf("Name:Udeet Mittal\nBatch:C3\nRoll Number:64\n");
 int m,n,e,k;
NODE h1,h2,h3,h4;
h1=(NODE)malloc(sizeof(struct node));
h2=(NODE)malloc(sizeof(struct node));
h3=(NODE)malloc(sizeof(struct node));
h4=(NODE)malloc(sizeof(struct node));
h1->rlink=h1;
h1->llink=h1;
h2->rlink=h2;
h2->llink=h2;
h3->rlink=h3;
h3->llink=h3;
h4->rlink=h4;
h4->llink=h4;
printf("\nnumber of nodes in list1\n");
scanf("%d",&n);
while(n>0)
{ scanf("%d",&m);
scanf("%d",&e);
h1=add(h1,m,e);
n--;
}
display(h1);
printf("\nnumber of nodes in list2\n");
scanf("%d",&k);
while(k>0)
```

```
{ scanf("%d",&m);
scanf("%d",&e);
h2=add(h2,m,e);
k--;
}
display(h2);
printf("\nthe sum is\n");
h3=sum(h1,h2,h3);
display(h3);
return 1;
}
```



Questions for Lab7

1) Implement a queue using singly linked list without header node.

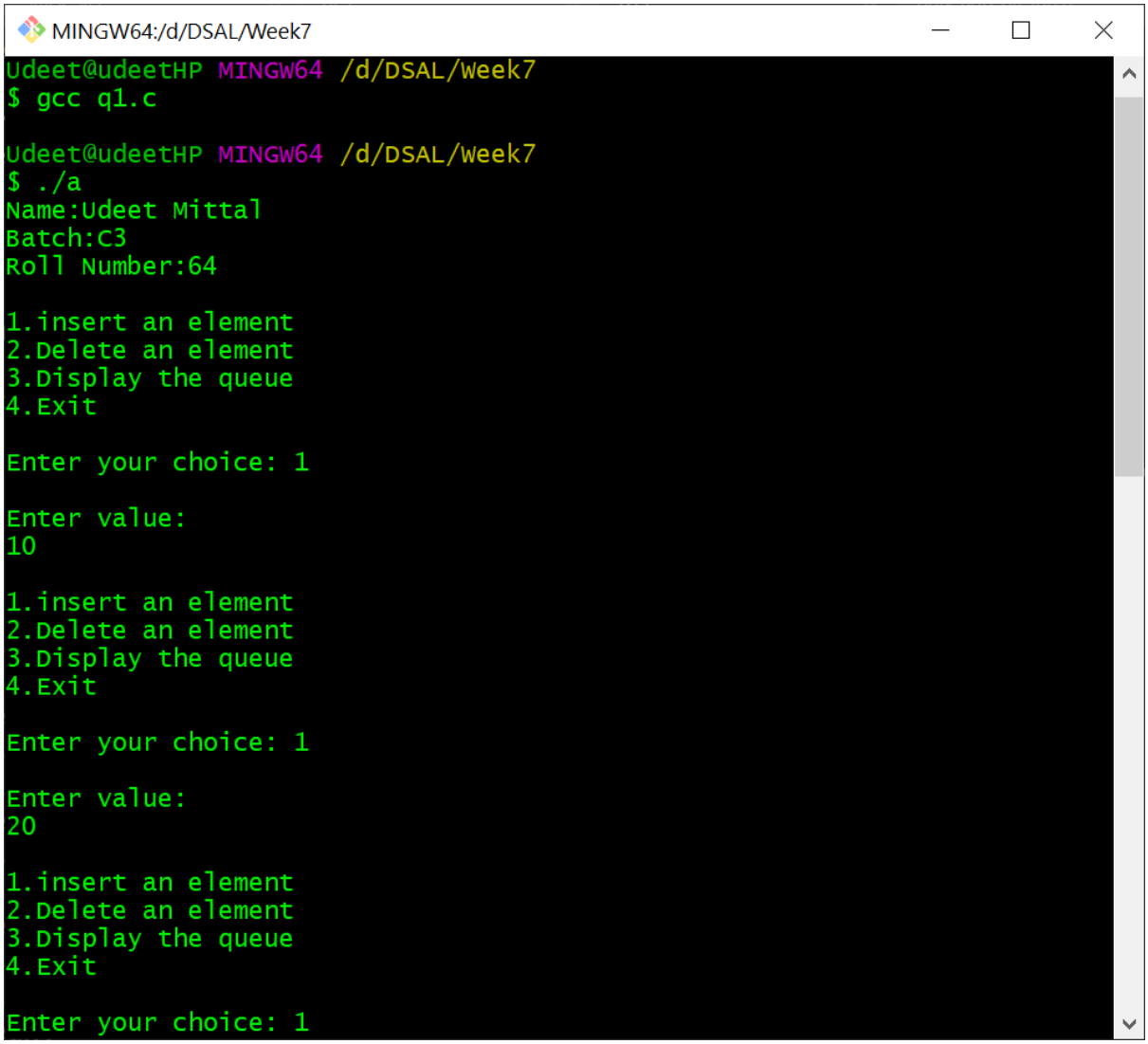
Filename: q1.c

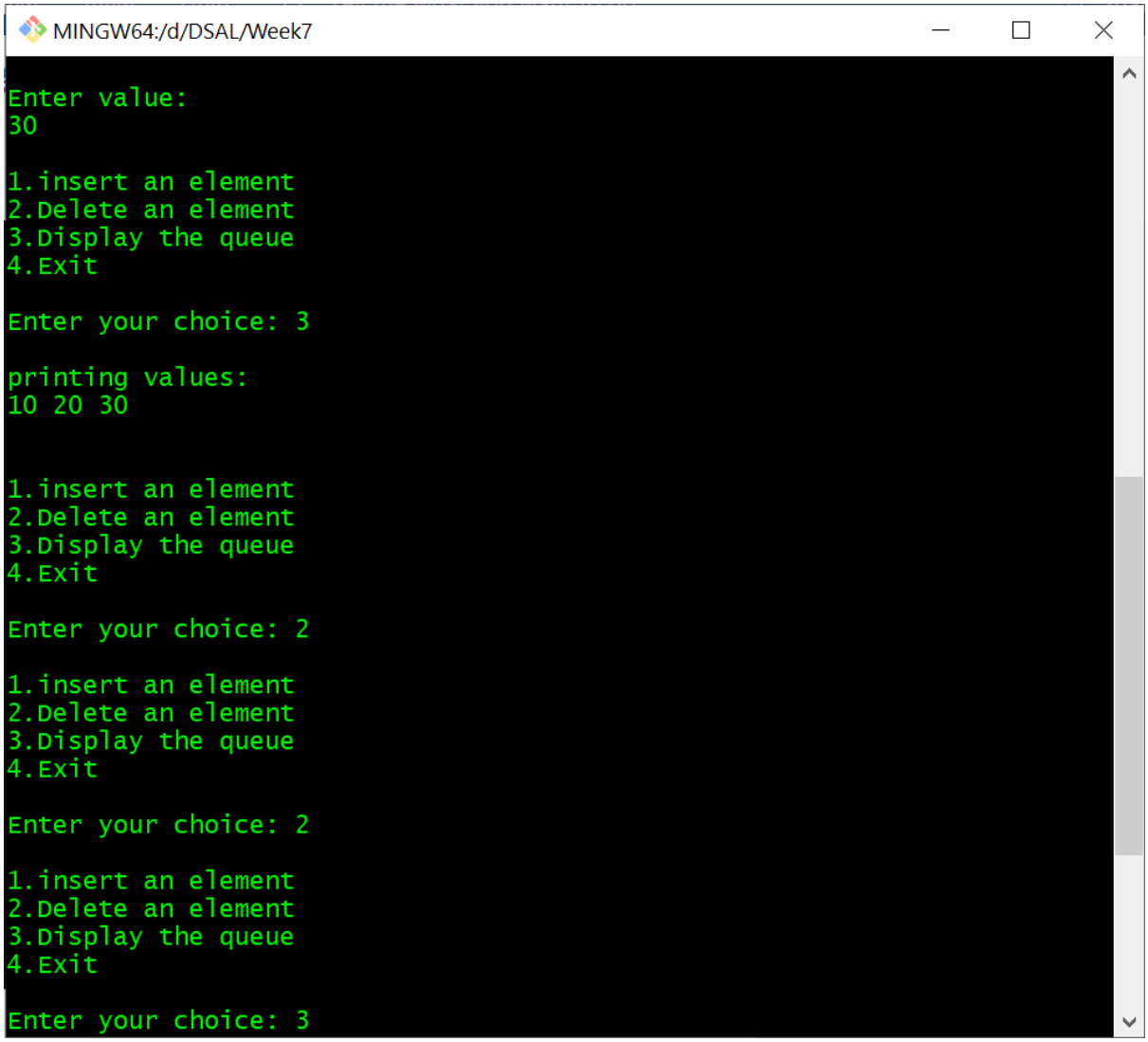
```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node *next;
};
struct node *front;
struct node *rear;
void insert();
void delete();
void display();
void main ()
{   printf("Name:Udeet Mittal\nBatch:C3\nRoll Number:64\n");
   int choice;
   while(choice != 4)
   {
      printf("\n1.insert an element\n2.Delete an element\n3.Display the
queue\n4.Exit\n");
      printf("\nEnter your choice: ");
      scanf("%d",&choice);
      switch(choice)
      {
         case 1:
```

```
            insert();
            break;
            case 2:
            delete();
            break;
            case 3:
            display();
            break;
            case 4:
            exit(0);
            break;
            default:
            printf("\nEnter valid choice\n");
        }
    }
}
void insert()
{
    struct node *ptr;
    int item;

    ptr = (struct node *) malloc (sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW\n");
        return;
    }
    else
    {
```

```
        printf("\nEnter value: \n");
        scanf("%d",&item);
        ptr -> data = item;
        if(front == NULL)
        {
            front = ptr;
            rear = ptr;
            front -> next = NULL;
            rear -> next = NULL;
        }
        else
        {
            rear -> next = ptr;
            rear = ptr;
            rear->next = NULL;
        }
    }
}
void delete ()
{
    struct node *ptr;
    if(front == NULL)
    {
        printf("\nUNDERFLOW\n");
        return;
    }
    else
    {
        ptr = front;
```

```
        front = front -> next;
        free(ptr);
    }
}
void display()
{
    struct node *ptr;
    ptr = front;
    if(front == NULL)
    {
        printf("\nEmpty queue\n");
    }
    else
    {   printf("\nprinting values: \n");
        while(ptr != NULL)
        {
            printf("%d ",ptr -> data);
            ptr = ptr -> next;
        }
        printf("\n\n");
    }
}
```


MINGW64:/d/DSAL/Week7
Udeet@udeetHP MINGW64 /d/DSAL/Week7
$ gcc q1.c
Udeet@udeetHP MINGW64 /d/DSAL/Week7
$ ./a
Name:Udeet Mittal
Batch:C3
Roll Number:64
1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice: 1
Enter value:
10
1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice: 1
Enter value:
20
1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice: 1



2) Perform UNION and INTERSECTION set operations on singly linked lists with header node.

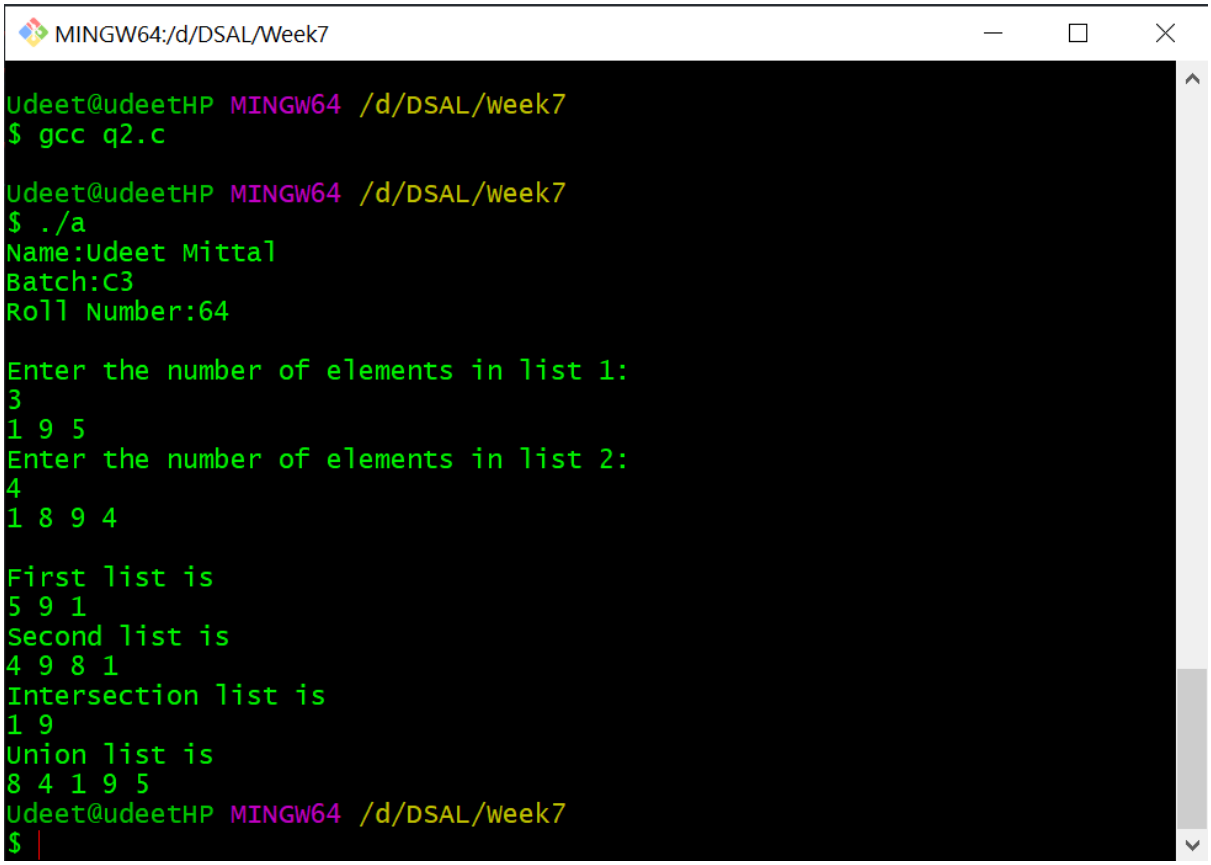
Filename: q2.c

```
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
struct Node {
	int data;
	struct Node* next;
};

void push(struct Node** head_ref, int new_data);
bool isPresent(struct Node* head, int data);
struct Node* getUnion(
	struct Node* head1,
	struct Node* head2)
{
	struct Node* result = NULL;
	struct Node *t1 = head1, *t2 = head2;
	while (t1 != NULL) {
		push(&result, t1->data);
		t1 = t1->next;
	}

	while (t2 != NULL) {
		if (!isPresent(result, t2->data))
			push(&result, t2->data);
		t2 = t2->next;
	}
```

```
        return result;
}
struct Node* getIntersection(struct Node* head1,
                                                    struct Node* head2)
{
        struct Node* result = NULL;
        struct Node* t1 = head1;

        while (t1 != NULL) {
                if (isPresent(head2, t1->data))
                        push(&result, t1->data);
                t1 = t1->next;
        }
        return result;
}
void push(struct Node** head_ref, int new_data)
{
        struct Node* new_node
                = (struct Node*) malloc(sizeof(struct Node));
        new_node->data = new_data;
        new_node->next = (*head_ref);
        (*head_ref) = new_node;
}

void printList(struct Node* node)
{
        while (node != NULL) {
                printf("%d ", node->data);
                node = node->next;
```

```
        }
}
bool isPresent(struct Node* head, int data)
{
        struct Node* t = head;
        while (t != NULL) {
                if (t->data == data)
                        return 1;
                t = t->next;
        }
        return 0;
}
int main()
{       struct Node* head1 = NULL;
        struct Node* head2 = NULL;
        struct Node* intersecn = NULL;
        struct Node* unin = NULL;
        printf("Name:Udeet Mittal\nBatch:C3\nRoll Number:64\n\n");
        printf("Enter the number of elements in list 1:\n");
        int n;scanf("%d",&n);
        for(int i=1;i<=n;i++)
        {int x;
                scanf("%d",&x);
        push(&head1,x);
        }

        printf("Enter the number of elements in list 2:\n");
        int m;scanf("%d",&m);
        for(int i=1;i<=m;i++)
```

```
        {int x;
                scanf("%d",&x);
        push(&head2,x);
        }
        intersecn = getIntersection(head1, head2);
        unin = getUnion(head1, head2);
        printf("\nFirst list is \n");
        printList(head1);
        printf("\nSecond list is \n");
        printList(head2);
        printf("\nIntersection list is \n");
        printList(intersecn);
        printf("\nUnion list is \n");
        printList(unin);
        return 0;
}
```


MINGW64:/d/DSAL/Week7
Udeet@udeetHP MINGW64 /d/DSAL/Week7
$ gcc q2.c
Udeet@udeetHP MINGW64 /d/DSAL/Week7
$ ./a
Name:Udeet Mittal
Batch:C3
Roll Number:64
Enter the number of elements in list 1:
3
1 9 5
Enter the number of elements in list 2:
4
1 8 9 4
First list is
5 9 1
Second list is
4 9 8 1
Intersection list is
1 9
Union list is
8 4 1 9 5
Udeet@udeetHP MINGW64 /d/DSAL/Week7
$