DSAL

Week 6

Udeet Mittal

CSE C3

Roll Number 64

1.SOLVED EXERCISE:
1) Implement a dequeue of integers with following functions.
a) deleteLeft b) addLeft c) deleteRight d) addRight e) display

filename: dequeue_fun.h

```
#define MAX 30
typedef struct dequeue
{
int data[MAX];
int rear,front;
}dequeue;

void initialize(dequeue *p);
int empty(dequeue *p);
int full(dequeue *p);
void enqueueR(dequeue *p,int x);
void enqueueF(dequeue *p,int x);
int dequeueF(dequeue *p);
int dequeueR(dequeue *p);
void print(dequeue *p);

void initialize(dequeue *P)
{
P->rear=-1;
P->front=-1;
}

int empty(dequeue *P)
{
if(P->rear==-1)
return(1);
return(0);
}
```

```
int full(dequeue *P)
{
if((P->rear+1)%MAX==P->front)
return(1);
return(0);
}

void enqueueR(dequeue *P,int x)
{
if(empty(P))
{
P->rear=0;
P->front=0;
P->data[0]=x;
}
else
{
P->rear=(P->rear+1)%MAX;
P->data[P->rear]=x;
}
}

void enqueueF(dequeue *P,int x)
{
if(empty(P))
{
P->rear=0;
P->front=0;
P->data[0]=x;
}
else
{
P->front=(P->front-1+MAX)%MAX;
P->data[P->front]=x;
}
}

int dequeueF(dequeue *P)
{
int x;
x=P->data[P->front];
if(P->rear==P->front) /*delete the last element */
initialize(P);
else
P->front=(P->front+1)%MAX;
return(x);
}
```

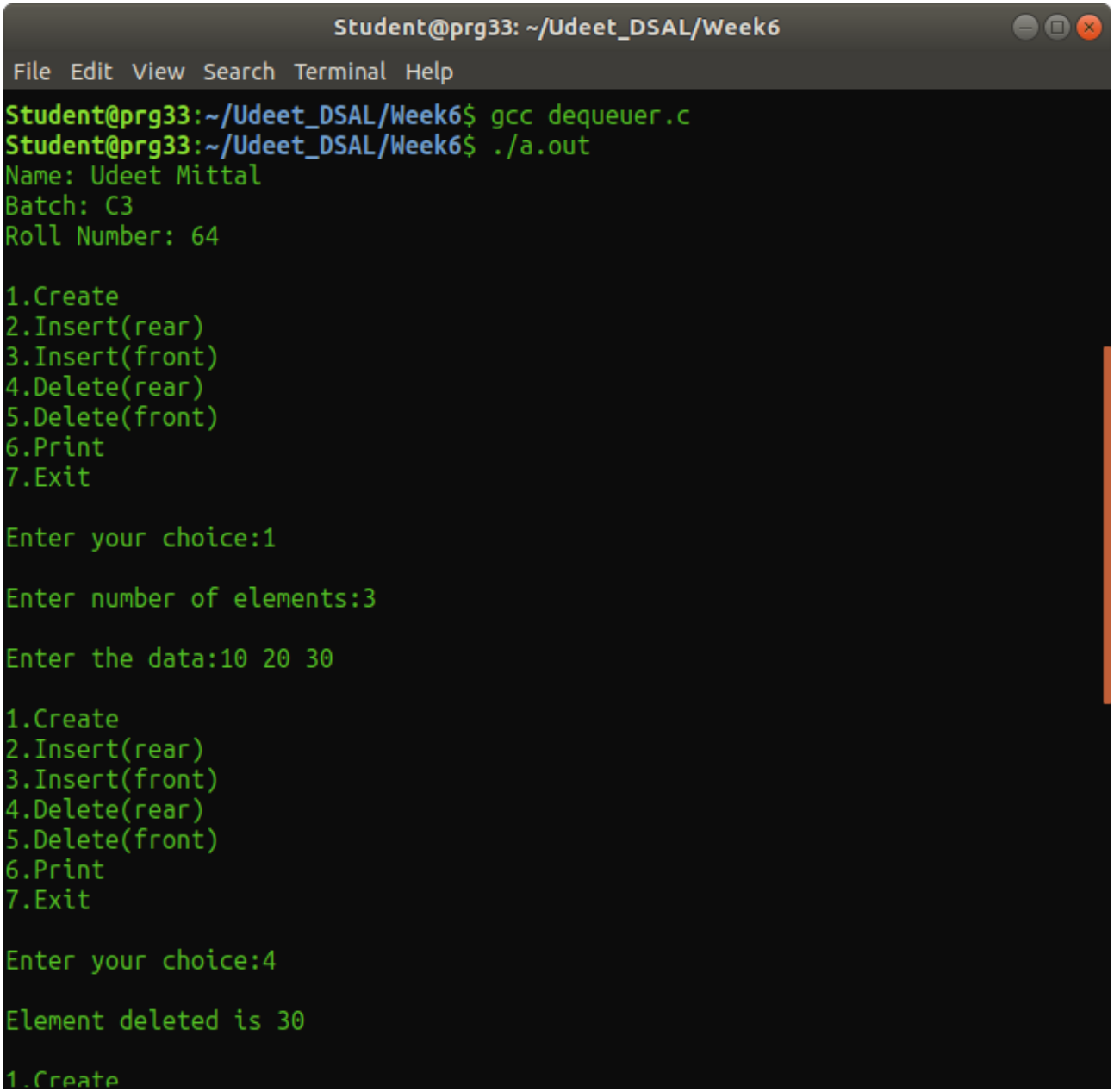
```
int dequeueR(dequeue *P)
{
int x;
x=P->data[P->rear];
if(P->rear==P->front)
initialize(P);
else
P->rear=(P->rear-1+MAX)%MAX;
return(x);
}

void print(dequeue *P)
{
if(empty(P))
{
printf("\nQueue is empty!!");
exit(0);
}
int i;
i=P->front;
while(i!=P->rear)
{
printf("\n%d",P->data[i]);
i=(i+1)%MAX;
}
printf("\n%d\n",P->data[P->rear]);
}
```

filename: dequeuer.c

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

int main()
{
printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
int i,x,op,n;
dequeue q;
initialize(&q);
do
{
printf("\n1.Create\n2.Insert(rear)\n3.Insert(front)\n4.Delete(rear)\n5.Delete(front)");
printf("\n6.Print\n7.Exit\n\nEnter your choice:");
scanf("%d",&op);
switch(op)
```

```
{

case 1: printf("\nEnter number of elements:");
scanf("%d",&n);
initialize(&q);
printf("\nEnter the data:");
for(i=0;i<n;i++)
{
scanf("%d",&x);
if(full(&q))
{
printf("\nQueue is full!!");
exit(0);
}
enqueueR(&q,x);
}
break;

case 2: printf("\nEnter element to be inserted:");
scanf("%d",&x);
if(full(&q))
{
printf("\nQueue is full!!");
exit(0);
}
enqueueR(&q,x);
break;

case 3: printf("\nEnter the element to be inserted:");
scanf("%d",&x);
if(full(&q))
{
printf("\nQueue is full!!");
exit(0);
}
enqueueF(&q,x);
break;

case 4: if(empty(&q))
{
printf("\nQueue is empty!!");
exit(0);
}
x=dequeueR(&q);
printf("\nElement deleted is %d\n",x);
break;

case 5: if(empty(&q))
```

```
{
printf("\nQueue is empty!!");
exit(0);
}
x=dequeueF(&q);
printf("\nElement deleted is %d\n",x);
break;

case 6: print(&q);
break;
default: break;
}
}while(op!=7);
return 0;
}
```



```
Student@prg33: ~/Udeet_DSAL/Week6
File  Edit  View  Search  Terminal  Help
1.Create
2.Insert(rear)
3.Insert(front)
4.Delete(rear)
5.Delete(front)
6.Print
7.Exit

Enter your choice:2

Enter element to be inserted:50

1.Create
2.Insert(rear)
3.Insert(front)
4.Delete(rear)
5.Delete(front)
6.Print
7.Exit

Enter your choice:6

10
20
50

1.Create
2.Insert(rear)
3.Insert(front)
4.Delete(rear)
5.Delete(front)
6.Print
7.Exit
```

```
Student@prg33: ~/Udeet_DSAL/Week6
File  Edit  View  Search  Terminal  Help

Enter your choice:7
Student@prg33:~/Udeet_DSAL/Week6$
```

Questions for Lab6

1) Implement an ascending priority queue.

Note: An ascending priority queue is a collection of items into which items can be inserted arbitrarily and from which only the smallest item can be removed. If apq is an ascending priority queue, the operation pqinsert(apq,x) inserts element x into apq and pqmindelete(apq) removes the minimum element from apq and returns its value

filename: priorq.h

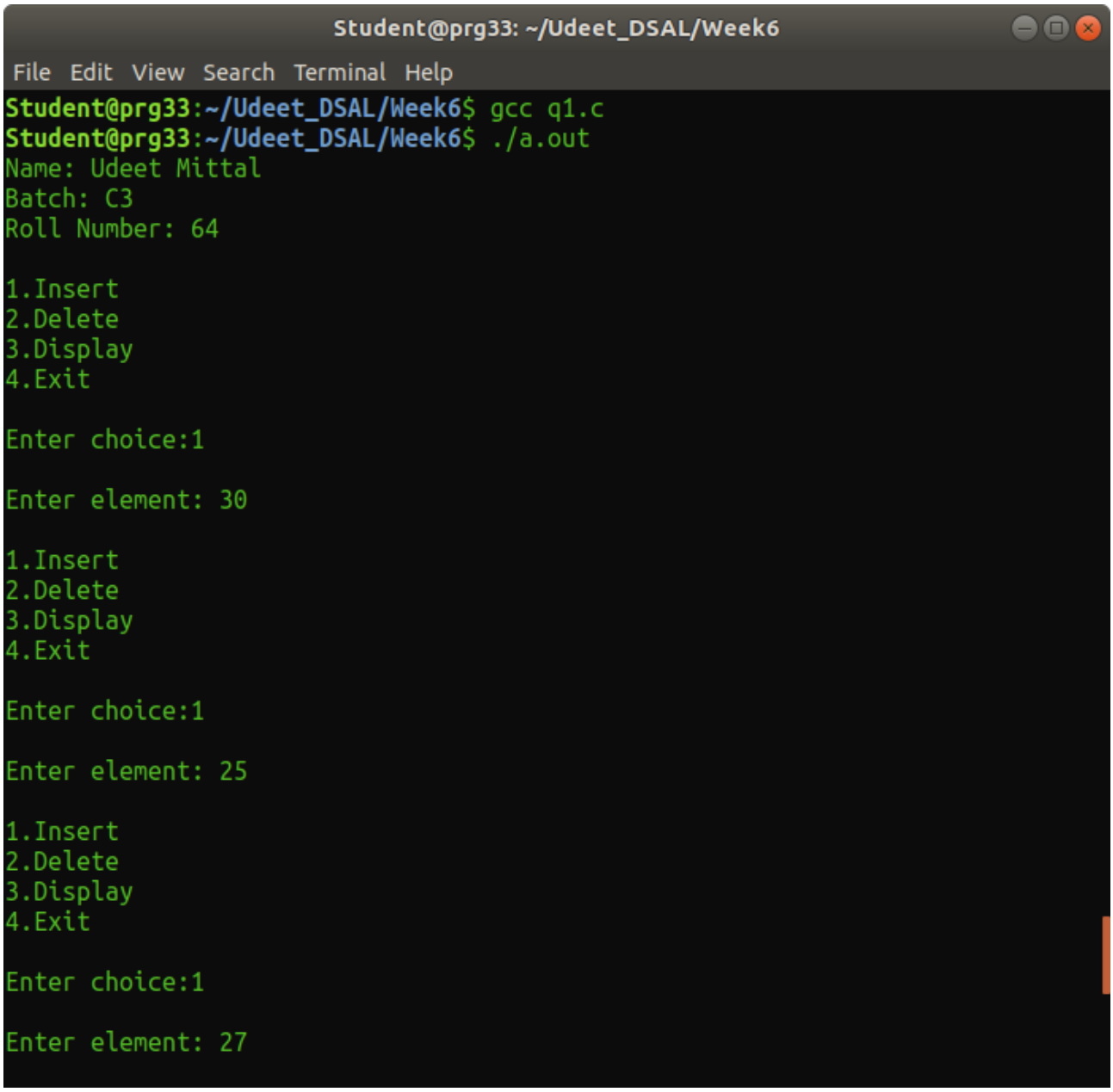
```
#define MAX 30
#include<stdio.h>

typedef struct {
int data[MAX];
int front,rear;
} pqueue ;

void initialise(pqueue * pq){
pq->rear=0;
}

void pqinsert(pqueue * pq,int x)
{
if(pq->rear==MAX)
{
printf("Overflow\n");
return;
}
pq->data[pq->rear++]=x;
}

int pqmindelete(pqueue *pq){
if(pq->rear==0)
{
printf("Underflow\n");
return -1;
}
int mi=0,mindata=0;
for(int i=0;i<pq->rear;i++)
{
if(pq->data[i]<pq->data[mi]){
mi=i;
mindata=pq->data[mi];
```

```
}
}
for(int i=mi;i<pq->rear-1;i++)
{
pq->data[i]=pq->data[i+1];
}
pq->rear--;
return mindata;
}

void display(pqueue * pq)
{
for(int i=0;i<pq->rear;i++)
{
printf("%d ",pq->data[i]);
}
printf("\n");
}
```

filename:q1.c

```
#include "priorq.h"
#include<stdio.h>
int main()
{
pqueue s;
pqueue *pq=&s;
initialise(pq);
int choice=0;
printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
while(choice!=4)
{
printf("\n1.Insert\n2.Delete\n3.Display\n4.Exit\n\nEnter choice:");
scanf("%d",&choice);
switch(choice)
{
case 1: printf("\nEnter element: ");
int x;
scanf("%d",&x);
pqinsert(pq,x);
break;
case 2:printf("%d popped\n",pqmindelete(pq));
break;
case 3: display(pq);
break;
case 4:break;
```

```
default:printf("Wrong choice \n");}
}
}
```



```
Student@prg33: ~/Udeet_DSAL/Week6
File  Edit  View  Search  Terminal  Help
1.Insert
2.Delete
3.Display
4.Exit

Enter choice:3
30 25 27

1.Insert
2.Delete
3.Display
4.Exit

Enter choice:2
25 popped

1.Insert
2.Delete
3.Display
4.Exit

Enter choice:2
27 popped

1.Insert
2.Delete
3.Display
4.Exit

Enter choice:3
30

1.Insert
```

```
Student@prg33: ~/Udeet_DSAL/Week6
File  Edit  View  Search  Terminal  Help
4.Exit

Enter choice:3
30

1.Insert
2.Delete
3.Display
4.Exit

Enter choice:4
Student@prg33:~/Udeet_DSAL/Week6$
```

2. Implement a queue of strings using an output restricted dequeue (no deleteRight).

Note: An output-restricted deque is one where insertion can be made at both ends, but deletion can be made from one end only, where as An input-restricted deque is one where deletion can be made from both ends, but insertion can be made at one end only

filename: q2.c

```
#include <stdio.h>
#include<stdbool.h>
#include<stdlib.h>
#define MAX 5

typedef struct dequeue{
char **data;
int rear,front;
}dequeue;

void initialize(dequeue *P){
P->rear=-1;P->front=-1;
}

bool empty(dequeue *P){
return(P->rear==-1);
}

bool full(dequeue *P){
return((P->rear+1)%MAX==P->front);
}

void enqueueR(dequeue *P){
char *x=(char*)calloc(25,sizeof(char));
printf("\nEnter the element to be inserted:");
scanf("%s",x);
if(empty(P)){
P->rear=0;
P->front=0;
P->data[0]=x;
} else{
P->rear=(P->rear+1)%MAX;
P->data[P->rear]=x;
}
```

```
}

void enqueueF(dequeue *P){
char *x=(char*)calloc(25,sizeof(char));
printf("\nEnter the element to be inserted:");
scanf("%s",x);
if(empty(P)){
P->rear=0;
P->front=0;
P->data[0]=x;
} else{
P->front=(P->front-1+MAX)%MAX;
P->data[P->front]=x;
}
}

char* dequeueF(dequeue *P)
{
char *x;
x=P->data[P->front];
if(P->rear==P->front)
/*delete the last element */
initialize(P);else
P->front=(P->front+1)%MAX;
return(x);
}

void print(dequeue *P){
if(empty(P)){
printf("\nQueue is empty!!");
return;
} int i=P->front;
while(i!=P->rear){
printf("\n%s",P->data[i]);
i=(i+1)%MAX;
}
printf("\n%s\n",P->data[P->rear]);
}

int main(){
printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
int i,op;
dequeue ptr;
dequeue *q=&ptr;
initialize(q);
q->data=(char**)calloc(MAX,sizeof(char*));
do{
printf("\n1.Insert(rear)\n2.Insert(front)\n3.Delete(front)");
```

```
printf("\n4.Print\n5.Exit\nEnter your choice:");
scanf("%d",&op);
switch(op)
{
case 1: if(full(q)){
printf("\nQueue is full!!");
break;
} enqueueR(q);
break;

case 2: if(full(q)){
printf("\nQueue is full!!");
break;
} enqueueF(q);
break;

case 3: if(empty(q)){
printf("\nQueue is empty!!");
break;}
printf("\nElement deleted is %s\n",dequeueF(q));
break;

case 4: print(q);
}
}while(op!=5);
return 0;
}
```

MINGW64:/d/DSAL/Week6

```
Udeet@udeetHP MINGW64 /d/DSAL/Week6
$ gcc q2.c

Udeet@udeetHP MINGW64 /d/DSAL/Week6
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:1

Enter the element to be inserted:10

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:1

Enter the element to be inserted:20

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:1

Enter the element to be inserted:30

1.Insert(rear)
```

MINGW64:/d/DSAL/Week6

```
Enter the element to be inserted:30

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:4

10
20
30

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:3

Element deleted is 10

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:2

Enter the element to be inserted:40

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:4
```

MINGW64:/d/DSAL/Week6

```
4.Print
5.Exit
Enter your choice:4

40
20
30

1.Insert(rear)
2.Insert(front)
3.Delete(front)
4.Print
5.Exit
Enter your choice:5

Udeet@udeetHP MINGW64 /d/DSAL/Week6
$
```

3. Write a program to check whether given string is a palindrome using a dequeue.

Filename: q3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 10

typedef struct
{ int front, rear;
        char array[MAX];
} Queue;

void pushR(Queue *q, char key)
{ if (q->rear == MAX -1)
{
printf("\nThe queue is full");
}
else
{ if (q->front == -1 && q->rear == -1)
{
q->front++;
}
q->array[++q->rear] = key;
}
}

char popRight(Queue *q)
{ char temp =q->array[q->rear];
q->rear--;
if (q->front > q->rear)
{
q->front = -1;
q->rear = -1;
}
return temp;
}

char popLeft(Queue *q)
{ char temp =q->array[q->front];
q->front++;
```

```
if (q->front > q->rear)
{q->front = -1;
q->rear = -1;
}
return temp;
}

int main()
{
printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
Queue ptr;
Queue *q=&ptr;
q->front = q->rear = -1;
char ele[100];
printf("\nEnter a string:");
scanf(" %s", ele);
int n = strlen(ele);
for(int i = 0; i<n; i++){
pushR(q, ele[i]);
}
n = n/2;
int p = 1;
while(n--)
{
if(popLeft(q)!=popRight(q)){
p = 0;
break;
}
}
if(p)
{
printf("%s is a Palindrome\n",ele);
}
else
{
printf("%s is not a Palindrome\n",ele);
}
return 0;
}
```

MINGW64:/d/DSAL/Week6

```
Udeet@udeetHP MINGW64 /d/DSAL/Week6
$ gcc q3.c

Udeet@udeetHP MINGW64 /d/DSAL/Week6
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter a string:hello
hello is not a Palindrome

Udeet@udeetHP MINGW64 /d/DSAL/Week6
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter a string:2112
2112 is a Palindrome

Udeet@udeetHP MINGW64 /d/DSAL/Week6
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter a string:racecar
racecar is a Palindrome

Udeet@udeetHP MINGW64 /d/DSAL/Week6
$
```