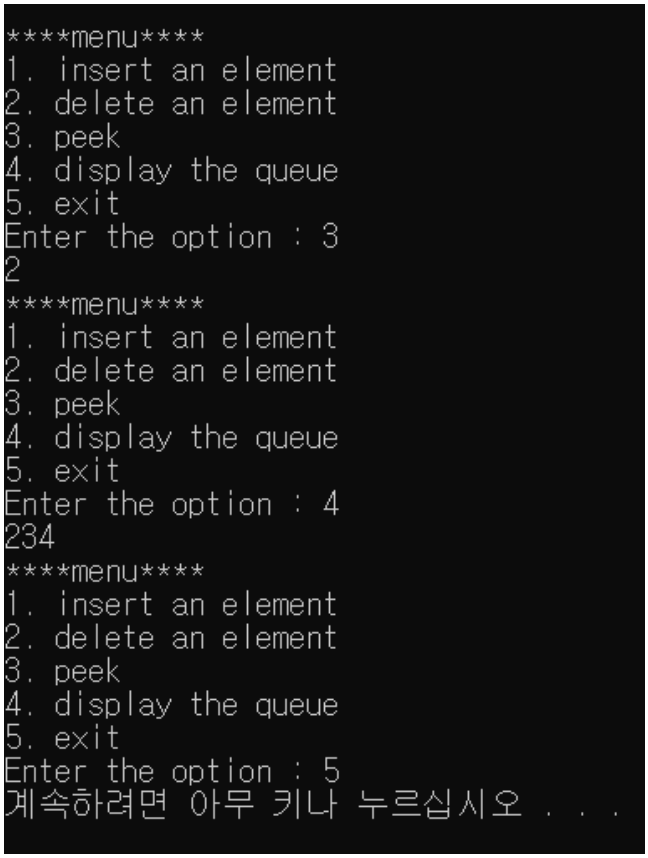
1.

```
#include <stdio.h>
#define max 10
int que[max];
int front=-1;
int rear=-1;
void insert();
void delete_();
void display();
int peek();
int main()
{
        int option, val;
        do
        {
        printf("\n");
        printf("****menu****\n");
        printf("1. insert an element\n");
        printf("2. delete an element\n");
        printf("3. peek\n");
        printf("4. display the queue\n");
        printf("5. exit\n");
        printf("Enter the option : ");
        scanf("%d",&option);
        switch(option)
        {
         case 1:
                    insert();
                    break;
         case 2:
                    delete_();
                    break;
         case 3:
                   val=peek();
                   printf("%d",val);
                   break;
         case 4:
                   display();
                   break;
        }


        }while(option!=5);
```

```
}

void insert()
{
        int num;
        printf("Enter the number to be inserted in the queue : ");
        scanf("%d",&num);
        if(rear==max-1)
                printf("overflow");
        else if(front==-1 && rear==-1)
                front=rear=0;
        else
                rear++;

        que[rear]=num;
}

void delete_()
{
        if(front==-1 || front>rear)
        {
                printf("underflow");
        }
        else
        {
                front++;
                if(front>rear)
                front=rear=-1;
        }
}

int peek()
{
        if(front==-1 || front>rear)
                printf("empty");
        else
                return que[front];
}

void display()
{
        int i;
        if(front==-1 || front>rear)
```

```
                printf("empty");
        else
        {
                for(i=front;  i<=rear;i++)
                printf("%d",que[i]);
        }
}
```

```
****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 1
Enter the number to be inserted in the queue : 1

****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 1
Enter the number to be inserted in the queue : 2

****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 1
Enter the number to be inserted in the queue : 3

****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 1
Enter the number to be inserted in the queue : 4

****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 2
```


****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 3
2
****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 4
234
****menu****
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter the option : 5
계속하려면 아무 키나 누르십시오 . . .

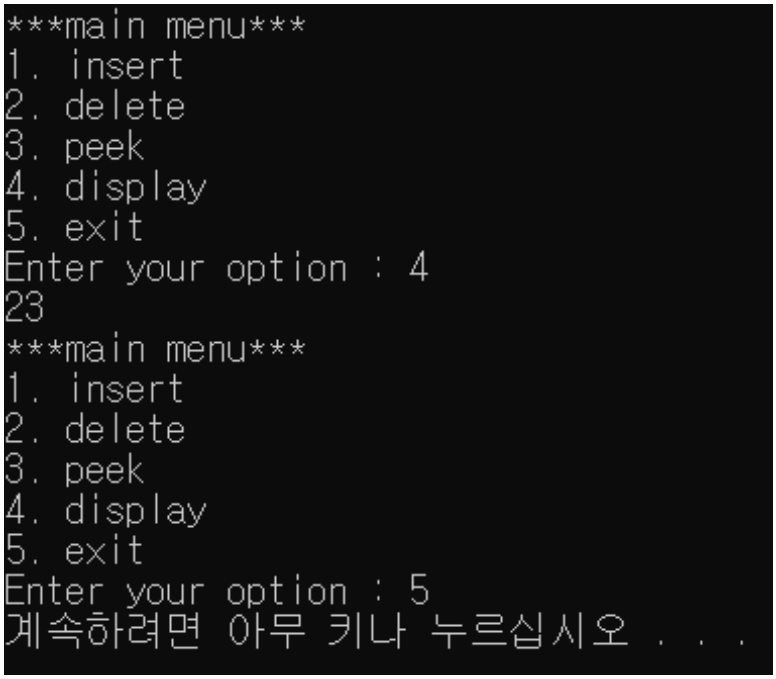
2.

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

struct node
{
        int data;
        struct node *next;
};
struct queue
{
        struct node *front;
        struct node *rear;
};
struct queue *q;
struct queue *create(struct queue *q);
struct queue *insert(struct queue *q, int val);
struct queue *delete_queue(struct queue *q);
void display(struct queue *q);
void peek(struct queue *p);
int main()
{
        int val,option;
        q=create(q);
        do
        {
                printf("\n");
                printf("***main menu***\n");
                printf("1. insert\n");
                printf("2. delete\n");
                printf("3. peek\n");
                printf("4. display\n");
                printf("5. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                        printf("Enter the number to insert in the queue : ");
                        scanf("%d",&val);
                        q=insert(q,val);
                        break;
```

```
                case 2:
                        q=delete_queue(q);
                        break;

                case 3:
                        peek(q);
                        break;
                case 4: display(q);
                        break;
                }

        }while(option!=5);
}

struct queue *create(struct queue *q)
{
        q=(struct queue*)malloc(sizeof(struct queue));
        q->front=NULL;
        q->rear=NULL;
        return q;
}

struct queue *insert(struct queue *q, int val)
{
        struct node *ptr;
        ptr=(struct node*)malloc(sizeof(struct node));
        ptr->data=val;
        if(q->front==NULL)
        {
                q->front=ptr;
                q->rear=ptr;
                q->front->next=q->rear->next=NULL;
        }
        else
        {
                q->rear->next=ptr;
                q->rear=ptr;
                q->rear->next=NULL;
        }
        return q;
}

struct queue *delete_queue(struct queue *q)
```

```
{
        struct node *ptr;
        ptr=q->front;
        if(q->front==NULL)
                printf("underflow");

        else
                {
                        q->front=q->front->next;
                        free(ptr);
                }
        return q;
}

void display(struct queue *q)
{
        struct node *ptr;
        ptr=q->front;
        if(ptr==NULL)
                printf("empty");
        else
        {
                while(ptr!=q->rear)
                {
                        printf("%d",ptr->data);
                        ptr=ptr->next;
                }
                printf("%d",ptr->data);
        }
}

void peek(struct queue *p)
{
        if(q->front==NULL)
                printf("empty");
        else
        {
                printf("%d",q->front->data);
        }
}
```

```
***main menu***
1. insert
2. delete
3. peek
4. display
5. exit
Enter your option : 1
Enter the number to insert in the queue : 1

***main menu***
1. insert
2. delete
3. peek
4. display
5. exit
Enter your option : 1
Enter the number to insert in the queue : 2

***main menu***
1. insert
2. delete
3. peek
4. display
5. exit
Enter your option : 1
Enter the number to insert in the queue : 3

***main menu***
1. insert
2. delete
3. peek
4. display
5. exit
Enter your option : 2

***main menu***
1. insert
2. delete
3. peek
4. display
5. exit
Enter your option : 3
2
```



3.

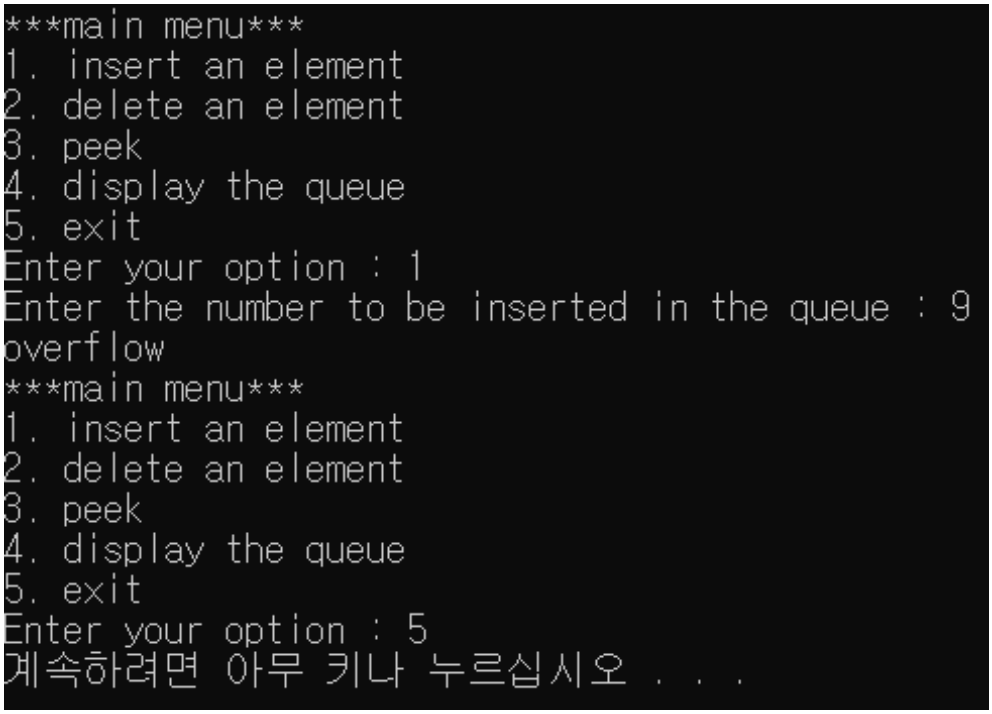
```
#include <stdio.h>
# define max 5
int queue[max];
int front=-1,rear=-1;
void insert();
void delete_element();
void peek();
void display();
int main()
{
        int option,val;
        do
        {
                printf("\n");
                printf("***main menu***\n");
                printf("1. insert an element\n");
                printf("2. delete an element\n");
                printf("3. peek\n");
                printf("4. display the queue\n");
                printf("5. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1: insert();
```

```
                        break;
                case 2: delete_element();
                        break;
                case 3: peek();
                        break;
                case 4: display();
                        break;
                }


        }while(option!=5);

}

void insert()
{
        int num;
        printf("Enter the number to be inserted in the queue : ");
        scanf("%d",&num);
        if((front==0 && rear==max-1)|| (front-rear==1))
        printf("overflow");
        else if(front==-1 && rear==-1)
        {
        front=rear=0;
        queue[rear]=num;
        }
        else if(rear==max-1 && front!=0)
        {
        rear=0;
        queue[rear]=num;
        }
        else
        {
        rear++;
        queue[rear]=num;
        }

}

void delete_element()
{
        int val;
        if(front==-1 && rear==-1)
```

```
        printf("underflow");
        if(front==rear)
                front=rear=-1;
        else
        {
                if(front==max-1)
                        front=0;
                else
                        front++;
        }

}

void peek()
{
        if(front==-1 && rear==-1)
                printf("empty");
        else
                printf("%d",queue[front]);
}

void display()
{
        int i;
        if(front==-1 && rear==-1)
                printf("empty");
        else
        {
                if(front<rear)
                {
                        for(i=front;i<=rear;i++)
                                printf("%d",queue[i]);
                }
                else
                {
                        for(i=front;i<max;i++)
                                printf("%d",queue[i]);
                        for(i=0;i<=rear;i++)
                                printf("%d",queue[i]);
                }
        }
}
```

```
***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 1
Enter the number to be inserted in the queue : 1

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 1
Enter the number to be inserted in the queue : 2

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 1
Enter the number to be inserted in the queue : 3

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 1
Enter the number to be inserted in the queue : 4

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 1
Enter the number to be inserted in the queue : 5
```

```
***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 2

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 2

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 1
Enter the number to be inserted in the queue : 7

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 1
Enter the number to be inserted in the queue : 8

***main menu***
1. insert an element
2. delete an element
3. peek
4. display the queue
5. exit
Enter your option : 4
34578
```



4.

```
#include <stdio.h>
#define max 5
int deque[max];
int left=-1;
int right=-1;
void insert_right();
void insert_left();
void delete_left();
void delete_right();
void display();
void output_deque();
void input_deque();
int main()
{
        int option;
        printf("****main menu****\n");
        printf("1. input restricted deque\n");
        printf("2. output restricted deque\n");
        printf("Enter your option : ");
        scanf("%d",&option);
        switch(option)
        {
                case 1: input_deque();
```

```
                        break;
                case 2: output_deque();
                        break;
        }
}

void input_deque()
{
        int option;
        do
        {
                printf("\n\n");
                printf("restricted deque\n");
                printf("1.insert at right\n");
                printf("2.delete from left\n");
                printf("3.delete from right\n");
                printf("4. display\n");
                printf("5. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                                insert_right();
                                break;
                case 2:
                                delete_left();
                                break;
                case 3:
                                delete_right();
                                break;
                case 4:
                        display();
                        break;
                }
        }while(option!=5);
}

void output_deque()
{
        int option;
        do
        {
```

```
                printf("\n\n");
                printf("restricted deque\n");
                printf("1.insert at right\n");
                printf("2.insert ar left\n");
                printf("3.delete from right\n");
                printf("4. display\n");
                printf("5. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                                insert_right();
                                break;
                case 2:
                                insert_left();
                                break;
                case 3:
                                delete_right();
                                break;
                case 4:
                        display();
                        break;
                }
        }while(option!=5);
}

void insert_right()
{
        int val;
        printf("Enter the value to be added : ");
        scanf("%d",&val);
        if((left==0 && right==max-1) || (left==right+1))
        {
                printf("overflow");
        }
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
```

```
                if(right==max-1)
                        right=0;
                else
                        right++;
        }
        deque[right]=val;
}

void insert_left()
{
        int val;
        printf("Enter the value to be added : ");
        scanf("%d",&val);
        if((left==0 && right ==max-1) || (left==right+1))
        {
                printf("overflow");
        }
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
                if(left==0)
                        left=max-1;
                else
                        left--;
        }
        deque[left]=val;
}

void delete_right()
{
        if(left==-1)
        {
                printf("underflow");
        }
        if(left==right)
        {
                left=-1;
                right=-1;
        }
```

```
        else
        {
                if(right==0)
                        right=max-1;
                else
                        right--;
        }
}

void delete_left()
{
        if(left==-1)
        {
                printf("underflow");
        }
        if(left==right)
        {
                left=-1;
                right=-1;
        }
        else
        {
                if(left==max-1)
                        left=0;
                else
                        left++;
        }
}

void display()
{
        int front=left, rear=right;
        if(front==-1)
        {
                printf("empty");
        }
        if(front<=rear)
        {
                while(front<=rear)
                {
                        printf("%d",deque[front]);
                        front++;
                }
```

```
		}
		else
		{
			while(front<=max-1)
			{
				printf("%d",deque[front]);
				front++;
			}
			front=0;
			while(front<=rear)
			{
				printf("%d",deque[front]);
				front++;
			}
		}
}
```

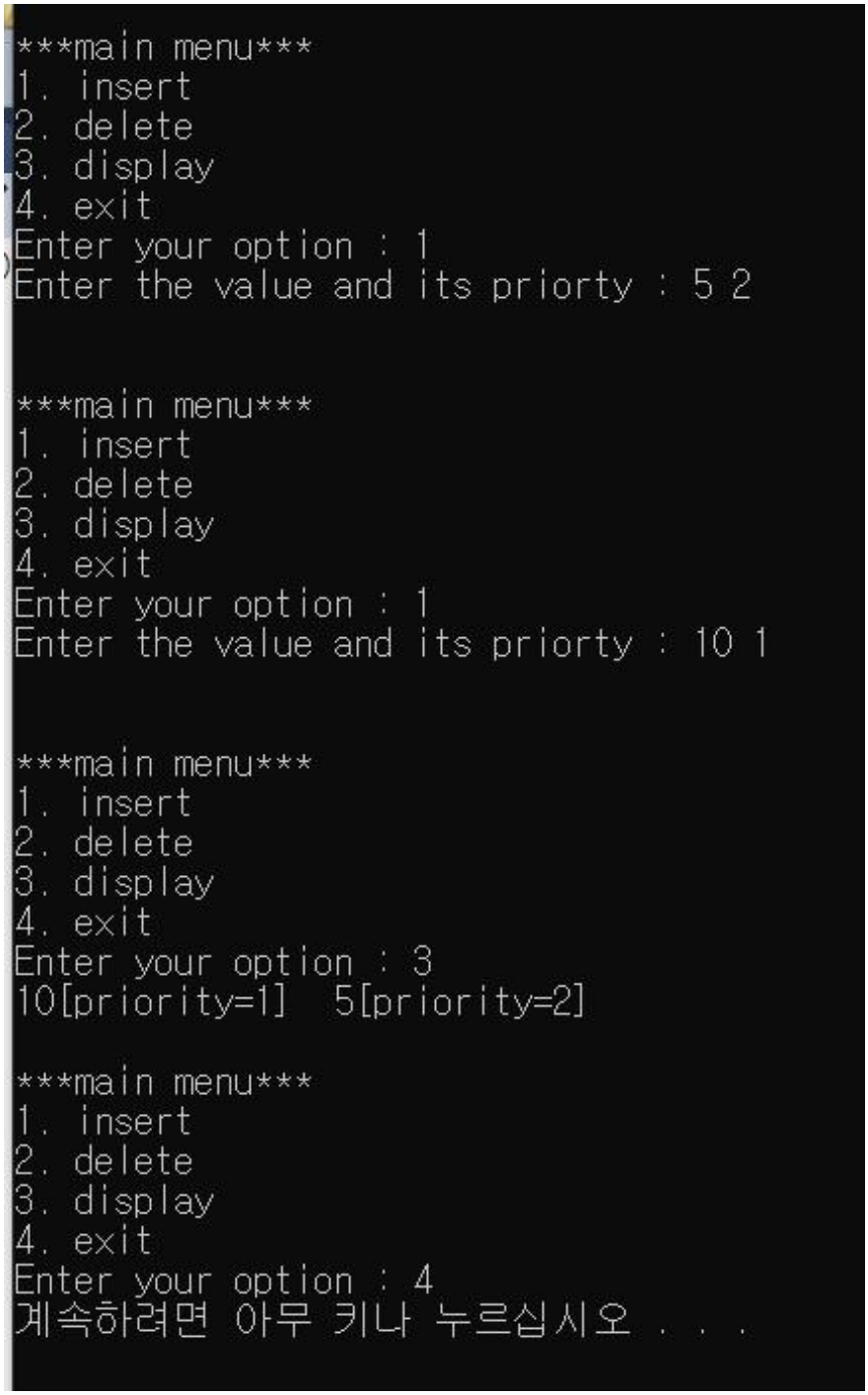
```
****main menu****
1. input restricted deque
2. output restricted deque
Enter your option : 1


restricted deque
1.insert at right
2.delete from left
3.delete from right
4. display
5. exit
Enter your option : 1
Enter the value to be added : 1


restricted deque
1.insert at right
2.delete from left
3.delete from right
4. display
5. exit
Enter your option : 1
Enter the value to be added : 2


restricted deque
1.insert at right
2.delete from left
3.delete from right
4. display
5. exit
Enter your option : 1
Enter the value to be added : 3


restricted deque
1.insert at right
2.delete from left
3.delete from right
4. display
5. exit
Enter your option : 4
123

restricted deque
1.insert at right
2.delete from left
3.delete from right
4. display
5. exit
Enter your option : 2


restricted deque
1.insert at right
2.delete from left
3.delete from right
4. display
5. exit
Enter your option : 3
```

5.

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


struct node
{
        int data;
        int prior;
        struct node *next;
};
struct node *start=NULL;
struct node *insert(struct node *start);
struct node *delete_node(struct node *start);
struct node *display(struct node *start);
int main()
{
        int option;
        do
        {
                printf("\n\n");
                printf("***main menu***\n");
                printf("1. insert\n");
                printf("2. delete\n");
                printf("3. display\n");
                printf("4. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
```

```
                switch(option)
                {
                case 1:
                        start=insert(start);
                        break;
                case 2:
                        start= delete_node(start);
                        break;
                case 3:
                        start= display(start);
                        break;

                }

        }while(option!=4);
}
struct node *insert(struct node *start)
{
        int val, pri;
        struct node *ptr,*p;
        ptr=(struct node *)malloc(sizeof(struct node));
        printf("Enter the value and its priorty : ");
        scanf("%d %d",&val,&pri);
        ptr->data=val;
        ptr->prior=pri;
        if(start==NULL || ptr->prior < start->prior)
        {
                ptr->next=start;
                start=ptr;
        }
        else
        {
                p=start;
                while(p->next!=NULL && p->next->prior<=ptr->prior)
                        p=p->next;
                ptr->next=p->next;
                p->next=ptr;
        }
        return start;
}

struct node *delete_node(struct node *start)
{
```

```
        struct node *ptr;
        if(start==NULL)
        {
                printf("underflow");
        }
        else
        {
                ptr=start;
                start=start->next;
                free(ptr);
        }
        return start;
}

struct node *display(struct node *start)
{
        struct node *ptr;
        ptr=start;
        if(start==NULL)
                printf("empty");
        else
        {
                while(ptr!=NULL)
                {
                        printf("%d[priority=%d]  ",ptr->data,ptr->prior);
                        ptr=ptr->next;
                }
        }
        return start;

}
```



6.

```
#include <stdio.h>
#define max 10
int queue[max];
int frontA=-1;
int rearA=-1;
int frontB=max;
int rearB=max;
```

```
void insertA(int val);
void insertB(int val);
void delete_a();
void delete_b();
void printA();
void printB();

int main()
{
        int option,val;
        do
        {       printf("\n");
                printf("******menu******\n");
                printf("1. insert in que A\n");
                printf("2. insert in que B\n");
                printf("3. delete from que A\n");
                printf("4. delete from que B\n");
                printf("5. display que A\n");
                printf("6. display que B\n");
                printf("7. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                                printf("Enter the data to be inserted in que A : ");
                                scanf("%d",&val);
                                insertA(val);
                                break;
                case 2:
                                printf("Enter the data to be inserted in que B : ");
                                scanf("%d",&val);
                                insertB(val);
                                break;
                case 3:
                                delete_a();
                                break;

                case 4:
                                delete_b();
                                break;
                case 5:
```

```
                                printA();
                                break;
                        case 6:
                                printB();
                                break;



                        }
                }while(option!=7);

}

void insertA(int val)
{
        if(rearA==rearB-1)
                printf("overflow");
        if(frontA==-1 && rearA==-1)
        {
                rearA=frontA=0;
                queue[rearA]=val;
        }
        else
        {
                rearA++;
                queue[rearA]=val;
        }
}

void insertB(int val)
{
        if(rearA==rearB-1)
                printf("overflow");
        if(frontB==max && rearB==max)
        {
                rearB=frontB=max-1;
                queue[rearB]=val;
        }
        else
        {
                rearB--;
                queue[rearB]=val;
        }
```

```
}

void delete_a()
{

	if(frontA==-1)
		printf("underflow");
	else
	{
		frontA++;
		if(frontA>rearA)
			frontA=rearA=-1;
	}
}

void delete_b()
{

	if(frontB==max)
		printf("underflow");
	else
	{
		frontB--;
		if(frontB<rearB)
			frontB=rearB=max;
	}
}

void printA()
{
	int i;
	if(frontA==-1)
		printf("empty");
	else
	{
	for(i=frontA;i<=rearA;i++)
		printf("%d",queue[i]);
	}
}

void printB()
{
	int i;
```

```
        if(frontB==max)
                printf("empty");
        else
        {
        for(i=frontB;i>=rearB;i--)
                printf("%d",queue[i]);
        }
}
```

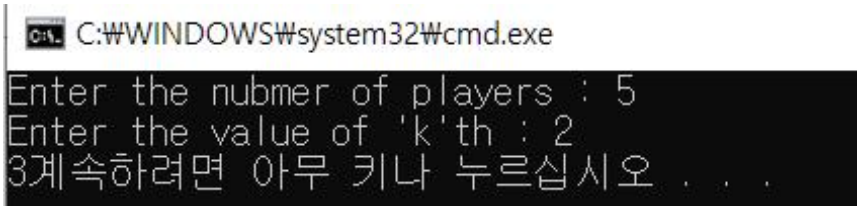
```
******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 1
Enter the data to be inserted in que A : 1

******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 1
Enter the data to be inserted in que A : 2

******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 1
Enter the data to be inserted in que A : 3

******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 2
Enter the data to be inserted in que B : 4

******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 2
Enter the data to be inserted in que B : 5
```

```
******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 2
Enter the data to be inserted in que B : 6

******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 5
123
******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 6
456
******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 3

******menu******
1. insert in que A
2. insert in que B
3. delete from que A
4. delete from que B
5. display que A
6. display que B
7. exit
Enter your option : 4
```

7.

```
#include <stdio.h>
#include <malloc.h>
struct node
{
        int id;
        struct node *next;
};
struct node *start=NULL;
void create(struct node *start, int num, int k);

int main()
{
        int num,k;
        printf("Enter the nubmer of players : ");
        scanf("%d",&num);
        printf("Enter the value of 'k'th");
        scanf("%d",&k);
```

```
        create(start,num,k);
}

void create(struct node *start, int num, int k)
{
        struct node *new_node,*ptr;
        int i,j;
        new_node=(struct node*)malloc(sizeof(struct node));
        new_node->id=1;
        new_node->next=NULL;
        start=new_node;
        ptr=start;
        for(i=2;i<=num;i++)
        {
                new_node=(struct node*)malloc(sizeof(struct node));
                ptr->next=new_node;
                new_node->id=i;
                new_node->next=start;
                ptr=new_node;
        }
        for(j=num;j>1;j--)
        {
                for(i=0;i<k-1;++i)
                        ptr=ptr->next;
                ptr->next=ptr->next->next;
        }
        printf("%d",ptr->id);


}
```



Programming Exercises

1.

```
#include <stdio.h>
#define max 5
int queue[max];
int front=-1;
int rear=-1;
int count=0;
void insert(int val);
void delete_element();
int main()
{
        int option,val;
        do
        {
                printf("\n");
                printf("***menu***\n");
                printf("1.insert\n");
                printf("2.delete\n");
                printf("3.calculate\n");
                printf("4.exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                        printf("Enter the number to be inserted in queue : ");
                        scanf("%d",&val);
                        insert(val);
                        break;
                case 2:
                        delete_element();
                        break;
                case 3:
                        printf("%d",count);
                        break;
                }
        }while(option!=4);
}

void insert(int val)
{
        if(rear==max-1)
```

```
                printf("overflow");
        else if(front==-1 && rear ==-1)
                front=rear=0;

        else
                rear++;
        count++;
        queue[rear]=val;

}

void delete_element()
{
        if(front==-1 || front>rear)
                printf("underflow");
        else
        {
                front++;
                if(front>rear)
                        front=rear=-1;
                count--;
        }
}
```

```
***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 1

***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 2

***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 3

***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 4

***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 2

***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 2
```

```
***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 3
2
***menu***
1.insert
2.delete
3.calculate
4.exit
Enter your option : 4
계속하려면 아무 키나 누르십시오 . . .
```

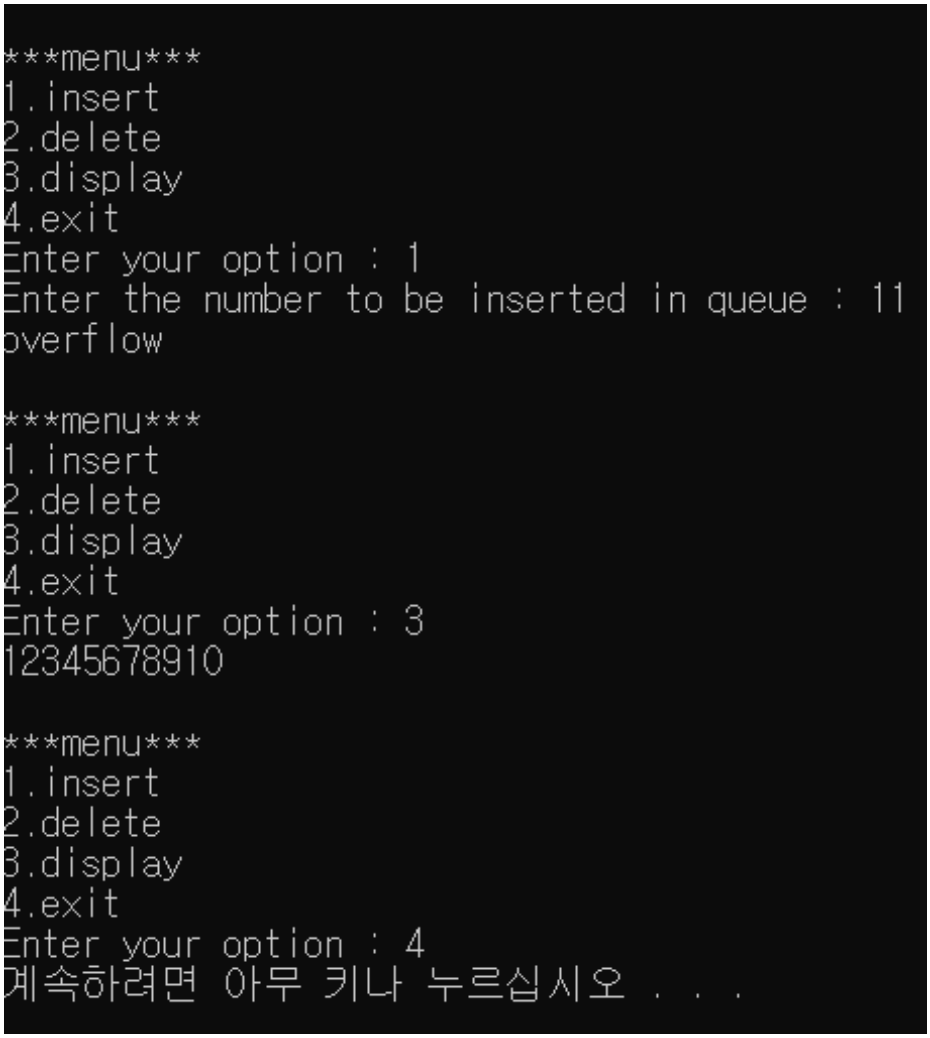
2.

```
#include <stdio.h>
#define max 10
int queue[max];
int front=-1;
int rear=-1;
void insert(int val);
void delete_element();
void display();
int main()
{
        int option,val;
        do
        {
                printf("\n\n");
                printf("***menu***\n");
                printf("1.insert\n");
                printf("2.delete\n");
                printf("3.display\n");
                printf("4.exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                        printf("Enter the number to be inserted in queue : ");
                        scanf("%d",&val);
                        insert(val);
                        break;
                case 2:
                        delete_element();
                        break;
                case 3:
                        display();
                        break;
                }
        }while(option!=4);
}

void insert(int val)
{
        if(rear==max-1)
                {
```

```
                        printf("overflow");
                        return;
                }
        else if(front==-1 && rear ==-1)
                front=rear=0;

        else
                rear++;
        queue[rear]=val;

}

void delete_element()
{
        if(front==-1 || front>rear)
        printf("underflow");
        else
        {
                front++;
                if(front>rear)
                        front=rear=-1;
        }
}
void display()
{
        int i;
        if(front==-1 || front > rear)
                printf("queue is empty");
        else
        {
                for(i=front;i<=rear;i++)
                        printf("%d",queue[i]);
        }
}
```

```
***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 1


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 2


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 3


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 4


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 5
```

```
***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 6


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 7


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 8


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 9


***menu***
1.insert
2.delete
3.display
4.exit
Enter your option : 1
Enter the number to be inserted in queue : 10
```



3.

```
#include <stdio.h>
# define max 10
int deque[max];
int left=-1;
int right=-1;
void insert_right();
void insert_left();
void display();

int main()
{
        int option,value;
        do{     printf("\n\n");
                printf("1. insert ar right\n");
                printf("2. insert at left\n");
                printf("3. display\n");
```

```
                printf("4. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1: insert_right();
                                break;
                case 2: insert_left();
                                break;
                case 3: display();
                                break;

                }
        }while(option!=4);
}

void insert_right()
{
        int val;
        printf("Enter the value to be added : ");
        scanf("%d",&val);
        printf("\n");
        if((left==0 && right==max-1) || (left==right+1))
                printf("overflow");
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
                if(right==max-1)
                        right=0;
                else
                        right++;
        }
        deque[right]=val;
}

void insert_left()
{
        int val;
        printf("Enter the value to be added : ");
```

```
        scanf("%d",&val);
        if((left==0 && right==max-1) || (left==right+1))
                printf("overflow");
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
                if(left==0)
                        left=max-1;
                else
                        left--;
        }
        deque[left]=val;
}

void display()
{
        int front=left;
        int rear=right;
        int i;
        if(front==-1)
                printf("empty");
        if(front<rear)
        {
                for(i=front;i<=rear;i++)
                        printf("%d",deque[i]);
        }
        else
        {
                for(i=front;i<=max-1;i++)
                        printf("%d",deque[i]);
                for(i=0;i<=rear;i++)
                        printf("%d",deque[i]);
        }

}
```

```
1. insert ar right
2. insert at left
3. display
4. exit
Enter your option : 1
Enter the value to be added : 1



1. insert ar right
2. insert at left
3. display
4. exit
Enter your option : 1
Enter the value to be added : 2



1. insert ar right
2. insert at left
3. display
4. exit
Enter your option : 1
Enter the value to be added : 3



1. insert ar right
2. insert at left
3. display
4. exit
Enter your option : 2
Enter the value to be added : 4


1. insert ar right
2. insert at left
3. display
4. exit
Enter your option : 3
4123

1. insert ar right
2. insert at left
3. display
4. exit
Enter your option : 4
계속하려면 아무 키나 누르십시오 . . .
```

4.

```
#include <stdio.h>
#include <malloc.h>
struct node
{
        int data;
        struct node *next;
};
struct queue
{
        struct node *front;
        struct node *rear;
};
struct queue *q;
struct queue *input_right(struct queue *q, int val);
struct queue *input_left(struct queue *q, int val);
struct queue* delete_left(struct queue *q);
struct queue* delete_right(struct queue *q);
struct queue* create_queue(struct queue *q);
struct queue* display(struct queue *q);
void input_deque(struct queue *q);
void output_deque(struct queue *q);
int main()
{
        int option;
        q=create_queue(q);
        printf("***main menu***\n");
        printf("1. input restricted deque\n");
        printf("2. delete restricted deque\n");
        printf("Enter your option : ");
        scanf("%d",&option);
        switch(option)
        {
        case 1: input_deque(q);
                        break;
        case 2: output_deque(q);
                        break;
        }
}

void input_deque(struct queue *q)
{
        int option,val;
```

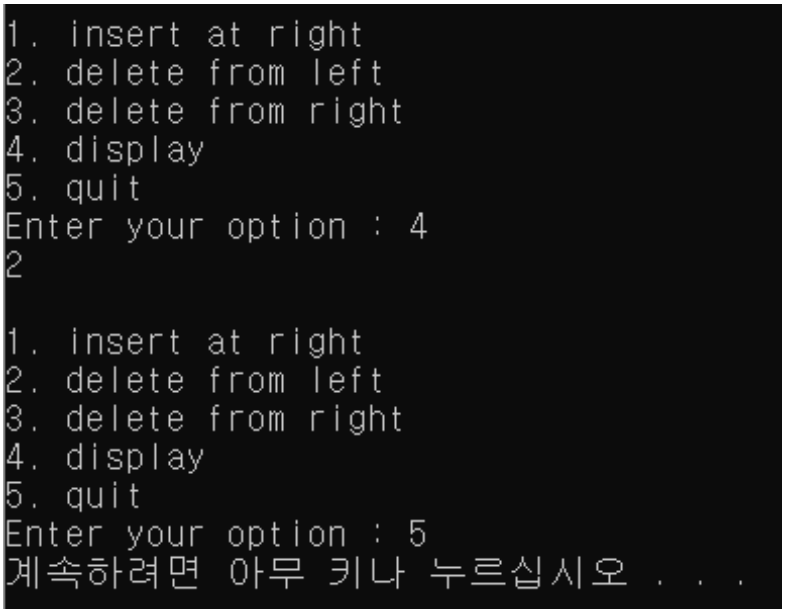
```
        do
        {
        printf("\n\n");
        printf("1. insert at right\n");
        printf("2. delete from left\n");
        printf("3. delete from right\n");
        printf("4. display\n");
        printf("5. quit\n");
        printf("Enter your option : ");
        scanf("%d",&option);
        switch(option)
        {
        case 1:  printf("Enter the number to be inserted in the que : ");
                 scanf("%d",&val);
                 q=input_right(q,val);
                 break;
        case 2: q=delete_left(q);
                 break;
        case 3: q=delete_right(q);
                 break;
        case 4: q=display(q);
                 break;

        }
        }while(option!=5);

}

void output_deque(struct queue *q)
{
        int option,val;
        do
        {
        printf("\n\n");
        printf("1. insert at right\n");
        printf("2. insert at left\n");
        printf("3. delete from left\n");
        printf("4. display\n");
        printf("5. quit\n");
        printf("Enter your option : ");
        scanf("%d",&option);
        switch(option)
        {
```

```
        case 1:  printf("Enter the number to be inserted in the que : ");
                        scanf("%d",&val);
                        q=input_right(q,val);
                        break;
        case 2: printf("Enter the number to be inserted in the que : ");
                        scanf("%d",&val);
                        q=input_left(q,val);
                        break;
        case 3: q=delete_left(q);
                        break;
        case 4: q=display(q);
                        break;

        }
        }while(option!=5);

}

struct queue *input_right(struct queue *q, int val)
{
        struct node *ptr;
        ptr=(struct node*)malloc(sizeof(struct node));
        ptr->data=val;
        if(q->front==NULL)
        {
                q->front=ptr;
                q->rear=ptr;
                q->front->next=q->rear->next=NULL;
        }
        else
        {
                q->rear->next=ptr;
                q->rear=ptr;
                q->rear->next=q->front;
        }
        return q;
}
struct queue *input_left(struct queue *q, int val)
{
        struct node *ptr;
        ptr=(struct node*)malloc(sizeof(struct node));
        ptr->data=val;
        if(q->front==NULL)
```

```
        {
                q->front=ptr;
                q->rear=ptr;
                q->front->next=q->rear->next=NULL;
        }
        else
        {
                ptr->next=q->front;
                q->front=ptr;
                q->rear->next=q->front;
        }
        return q;
}
struct queue* create_queue(struct queue *q)
{
        q=(struct queue*)malloc(sizeof(struct queue));
        q->rear=NULL;
        q->front=NULL;
        return q;
}

struct queue* display(struct queue *q)
{
        struct node *ptr;
        ptr=q->front;
        if(ptr==NULL)
                printf("empty");
        else
        {
                while(ptr->next!=q->front)
                {
                        printf("%d ",ptr->data);
                        ptr=ptr->next;
                }
                printf("%d",ptr->data);
        }
        return q;
}

struct queue* delete_left(struct queue *q)
{
        struct node *ptr;
        if(q->front==NULL)
```

```
                printf("underflow");
        else
        {
                ptr=q->front;
                q->front=q->front->next;
                q->rear->next=q->front;
                free(ptr);
        }
        return q;
}

struct queue* delete_right(struct queue *q)
{
        struct node *ptr,*preptr;
        ptr=q->front;
        if(q->front==NULL)
                printf("underflow");
        else
        {       while(ptr->next!=q->front)
                {
                        preptr=ptr;
                        ptr=ptr->next;
                }

                preptr->next=q->front;
                free(ptr);
        }
        return q;
}
```

```
***main menu***
1. input restricted deque
2. delete restricted deque
Enter your option : 1


1. insert at right
2. delete from left
3. delete from right
4. display
5. quit
Enter your option : 1
Enter the number to be inserted in the que : 1


1. insert at right
2. delete from left
3. delete from right
4. display
5. quit
Enter your option : 1
Enter the number to be inserted in the que : 2


1. insert at right
2. delete from left
3. delete from right
4. display
5. quit
Enter your option : 1
Enter the number to be inserted in the que : 3


1. insert at right
2. delete from left
3. delete from right
4. display
5. quit
Enter your option : 2


1. insert at right
2. delete from left
3. delete from right
4. display
5. quit
Enter your option : 3
```



5.

6.

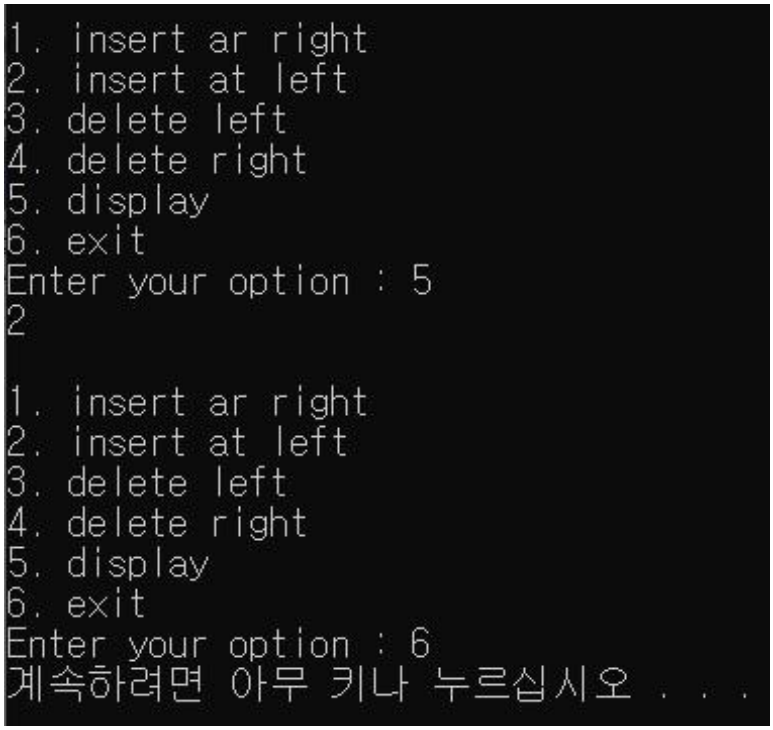
```
#include <stdio.h>
# define max 10
int deque[max];
int left=-1;
int right=-1;
void insert_right();
void insert_left();
void delete_right();
void delete_left();
void display();

int main()
{
        int option,value;
        do{     printf("\n\n");
                printf("1. insert ar right\n");
                printf("2. insert at left\n");
                printf("3. delete left\n");
                printf("4. delete right\n");
                printf("5. display\n");
                printf("6. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
```

```
                case 1: insert_right();
                                break;
                case 2: insert_left();
                                break;
                case 3: delete_left();
                                break;
                case 4: delete_right();
                                break;
                case 5: display();
                                break;

                }
        }while(option!=6);
}

void insert_right()
{
        int val;
        printf("Enter the value to be added : ");
        scanf("%d",&val);
        printf("\n");
        if((left==0 && right==max-1) || (left==right+1))
                printf("overflow");
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
                if(right==max-1)
                        right=0;
                else
                        right++;
        }
        deque[right]=val;
}

void insert_left()
{
        int val;
        printf("Enter the value to be added : ");
        scanf("%d",&val);
```

```
        if((left==0 && right==max-1) || (left==right+1))
                printf("overflow");
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
                if(left==0)
                        left=max-1;
                else
                        left--;
        }
        deque[left]=val;
}
void delete_left()
{
        if(left==-1)
        {
                printf("underflow");
                return;
        }
        if(left==right)
                left=right=-1;
        else
        {
                if(left==max-1)
                        left=0;
                else
                        left++;
        }
}
void delete_right()
{
        if(left==-1)
                {
                        printf("underflow");
                        return;
                }
        if(left==right)
                left=right=-1;
        else
```

```
        {
                if(right==0)
                        right=max-1;
                else
                        right--;
        }
}
void display()
{
        int front=left;
        int rear=right;
        int i;
        if(front==-1)
        {
                printf("empty");
                return;
        }
        if(front<=rear)
        {
                for(i=front;i<=rear;i++)
                        printf("%d",deque[i]);
        }
        else
        {
                for(i=front;i<=max-1;i++)
                        printf("%d",deque[i]);
                for(i=0;i<=rear;i++)
                        printf("%d",deque[i]);
        }

}
```

```
1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 1
Enter the value to be added : 1



1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 1
Enter the value to be added : 2



1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 1
Enter the value to be added : 3



1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 3


1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 4
```



7.

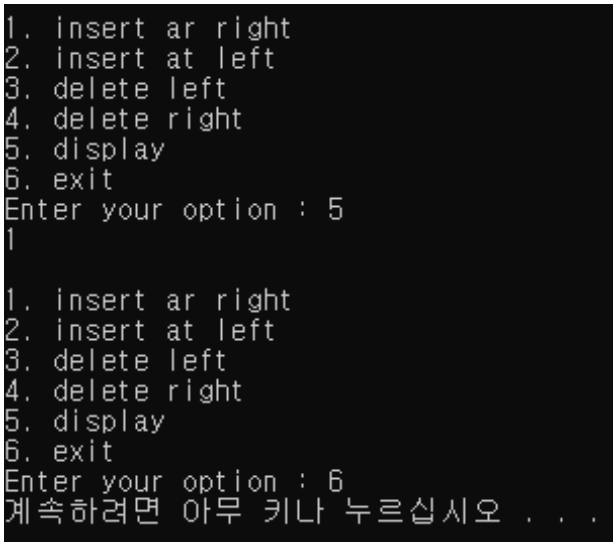
```
#include <stdio.h>
# define max 10
int deque[max];
int left=-1;
int right=-1;
void insert_right();
void insert_left();
void delete_right();
void delete_left();
void display();

int main()
{
        int option,value;
        do{     printf("\n\n");
                printf("1. insert ar right\n");
                printf("2. insert at left\n");
                printf("3. delete left\n");
                printf("4. delete right\n");
                printf("5. display\n");
                printf("6. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
```

```
                {
                case 1: insert_right();
                                break;
                case 2: insert_left();
                                break;
                case 3: delete_left();
                                break;
                case 4: delete_right();
                                break;
                case 5: display();
                                break;

                }
        }while(option!=6);
}

void insert_right()
{
        int val;
        printf("Enter the value to be added : ");
        scanf("%d",&val);
        printf("\n");
        if((left==0 && right==max-1) || (left==right+1))
                printf("overflow");
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
                if(right==max-1)
                        right=0;
                else
                        right++;
        }
        deque[right]=val;
}

void insert_left()
{
        int val;
        printf("Enter the value to be added : ");
```

```
        scanf("%d",&val);
        if((left==0 && right==max-1) || (left==right+1))
                printf("overflow");
        if(left==-1)
        {
                left=0;
                right=0;
        }
        else
        {
                if(left==0)
                        left=max-1;
                else
                        left--;
        }
        deque[left]=val;
}
void delete_left()
{
        if(left==-1)
        {
                printf("underflow");
                return;
        }
        if(left==right)
                left=right=-1;
        else
        {
                if(left==max-1)
                        left=0;
                else
                        left++;
        }
}
void delete_right()
{
        if(left==-1)
                {
                        printf("underflow");
                        return;
                }
        if(left==right)
                left=right=-1;
```

```
        else
        {
                if(right==0)
                        right=max-1;
                else
                        right--;
        }
}
void display()
{
        int front=left;
        int rear=right;
        int i;
        if(front==-1)
        {
                printf("empty");
                return;
        }
        if(front<=rear)
        {
                for(i=front;i<=rear;i++)
                        printf("%d",deque[i]);
        }
        else
        {
                for(i=front;i<=max-1;i++)
                        printf("%d",deque[i]);
                for(i=0;i<=rear;i++)
                        printf("%d",deque[i]);
        }

}
```

```
1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 1
Enter the value to be added : 1



1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 2
Enter the value to be added : 2


1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 1
Enter the value to be added : 3



1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 5
213

1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 3


1. insert ar right
2. insert at left
3. delete left
4. delete right
5. display
6. exit
Enter your option : 4
```

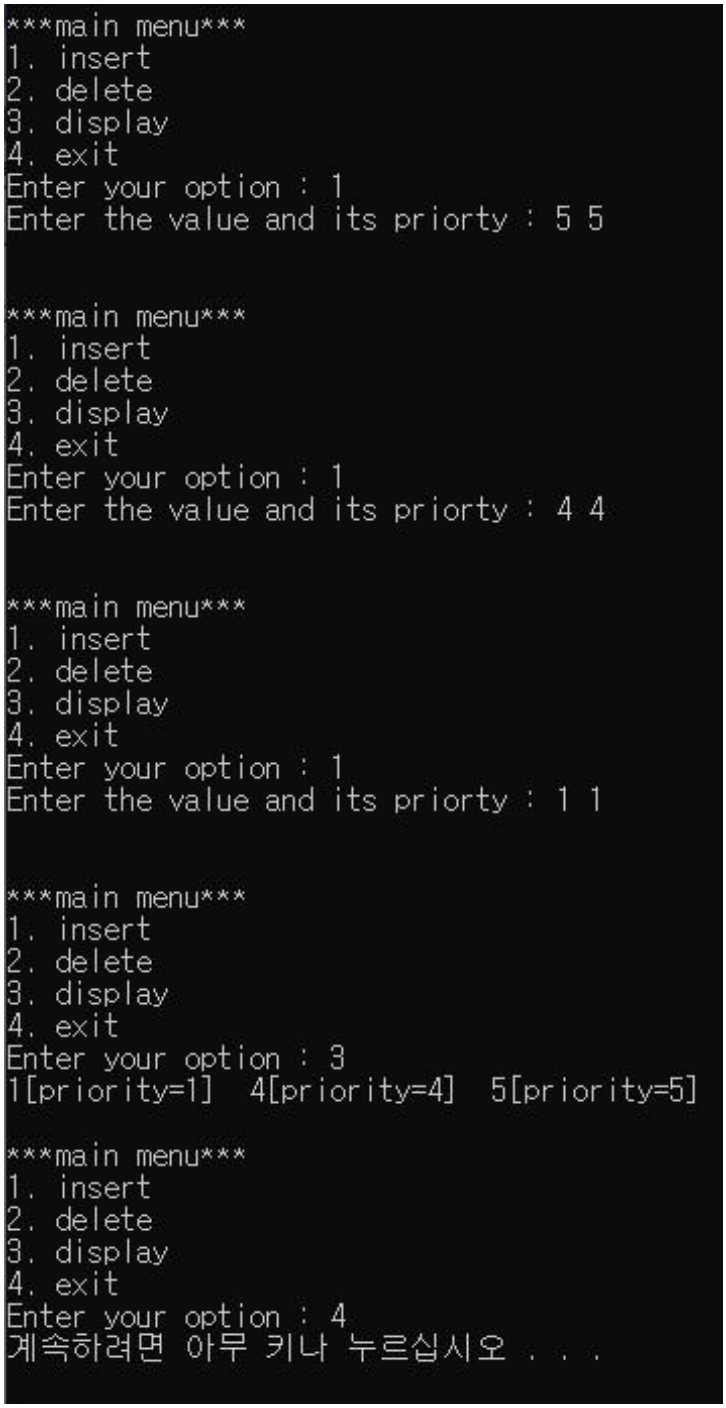


8.

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


struct node
{
        int data;
        int prior;
        struct node *next;
};
struct node *start=NULL;
struct node *insert(struct node *start);
struct node *delete_node(struct node *start);
struct node *display(struct node *start);
int main()
{
        int option;
        do
        {
                printf("\n\n");
                printf("***main menu***\n");
                printf("1. insert\n");
                printf("2. delete\n");
                printf("3. display\n");
                printf("4. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
```

```
                case 1:
                        start=insert(start);
                        break;
                case 2:
                        start= delete_node(start);
                        break;
                case 3:
                        start= display(start);
                        break;

                }

        }while(option!=4);
}
struct node *insert(struct node *start)
{
        int val, pri;
        struct node *ptr,*p;
        ptr=(struct node *)malloc(sizeof(struct node));
        printf("Enter the value and its priorty : ");
        scanf("%d %d",&val,&pri);
        ptr->data=val;
        ptr->prior=pri;
        if(start==NULL || ptr->prior < start->prior)
        {
                ptr->next=start;
                start=ptr;
        }
        else
        {
                p=start;
                while(p->next!=NULL && p->next->prior<=ptr->prior)
                        p=p->next;
                ptr->next=p->next;
                p->next=ptr;
        }
        return start;
}

struct node *delete_node(struct node *start)
{
        struct node *ptr;
        if(start==NULL)
```

```
        {
                printf("underflow");
        }
        else
        {
                ptr=start;
                start=start->next;
                free(ptr);
        }
        return start;
}

struct node *display(struct node *start)
{
        struct node *ptr;
        ptr=start;
        if(start==NULL)
                printf("empty");
        else
        {
                while(ptr!=NULL)
                {
                        printf("%d[priority=%d]   ",ptr->data,ptr->prior);
                        ptr=ptr->next;
                }
        }
        return start;

}
```



9.

```
#include <stdio.h>
#define max 10
int stack[max];
int que[max];
int top=-1;
int front=-1;
int rear=-1;
```

```
void input_stack(int stack[], int val);
void change(int stack[]);
void stack_input(int num);
void display();
int main()
{
	int option,val;
	int result;
	do
	{
		printf("\n\n");
		printf("1. input stack\n");
		printf("2. create a que from stack\n");
		printf("3. display\n");
		printf("4. exit\n");
		printf("Enter your option");
		scanf("%d",&option);
		switch(option)
		{
		case 1:
			printf("Enter the number to be inserted stack : ");
			scanf("%d",&val);
			input_stack(stack,val);
			break;
		case 2:
			change(stack);
			break;
		case 3:
			display();
			break;
		}
	}while(option!=4);
}

void input_stack(int stack[], int val)
{
	if(top==max-1)
		printf("overflow");
	else
	{
		top++;
		stack[top]=val;
	}
```

```
}

void change(int stack[])
{
	int val;
	if(top==-1)
		printf("underflow");
	else
	{
		while(top!=-1)
		{
		val=stack[top];
		stack_input(val);
		top=top-1;


		}
	}
}

void stack_input(int num)
{
	if(rear==max-1)
		printf("overflow");
	else if(front==-1 && rear ==-1)
		front=rear=0;
	else
		rear++;
	que[rear]=num;
}

void display()
{
	int i;
	if(front==-1 || front>rear)
		printf("empty");
	else
	{
		for(i=0;i<=rear;i++)
			printf("%d", que[i]);
	}
}
```

```
1. input stack
2. create a que from stack
3. display
4. exit
Enter your option1
Enter the number to be inserted stack : 1


1. input stack
2. create a que from stack
3. display
4. exit
Enter your option1
Enter the number to be inserted stack : 2


1. input stack
2. create a que from stack
3. display
4. exit
Enter your option1
Enter the number to be inserted stack : 3


1. input stack
2. create a que from stack
3. display
4. exit
Enter your option2


1. input stack
2. create a que from stack
3. display
4. exit
Enter your option3
321

1. input stack
2. create a que from stack
3. display
4. exit
Enter your option4
계속하려면 아무 키나 누르십시오 . . .
```

10.

```
#include <stdio.h>
#define max 10
int stack[max];
int que[max];
int top=-1;
int front=-1;
int rear=-1;
void input_que(int que[], int val);
void change(int que[]);
void stack_input(int stack[], int val);
void display();

int main()
{
        int option,val;
        int result;
        do
        {
                printf("\n\n");
                printf("1. input que\n");
                printf("2. create a stack from que\n");
                printf("3. display\n");
                printf("4. exit\n");
                printf("Enter your option");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                        printf("Enter the number to be inserted queue : ");
                        scanf("%d",&val);
                        input_que(que,val);
                        break;
                case 2:
                        change(que);
                        break;
                case 3:
                        display();
                        break;
                }
        }while(option!=4);
}
```

```
void input_que(int que[], int val)
{
	if(rear==max-1)
		printf("overflow");
	else if(front==-1 && rear==-1)
		front=rear=0;
	else
		rear++;
	que[rear]=val;
}

void change(int que[])
{
	int val;
	if(top==max-1)
	{
		printf("overflow");
	}
	else
	{
		while(rear!=-1)
		{
			val=que[rear];
			stack_input(stack,val);
			rear--;
		}
	}

}

void stack_input(int stack[], int val)
{
	if(top==max-1)
		printf("overflow");
	else
	{
		top++;
		stack[top]=val;
	}
}

void display()
{
```

```
        int i;
        if(top==-1)
                printf("empty");
        else
        {
                for(i=0;i<=top;i++)
                        printf("%d", stack[i]);
        }
}
```

```
1. input que
2. create a stack from que
3. display
4. exit
Enter your option1
Enter the number to be inserted queue : 1


1. input que
2. create a stack from que
3. display
4. exit
Enter your option1
Enter the number to be inserted queue : 2


1. input que
2. create a stack from que
3. display
4. exit
Enter your option1
Enter the number to be inserted queue : 3


1. input que
2. create a stack from que
3. display
4. exit
Enter your option3
empty

1. input que
2. create a stack from que
3. display
4. exit
Enter your option2


1. input que
2. create a stack from que
3. display
4. exit
Enter your option3
321

1. input que
2. create a stack from que
3. display
4. exit
Enter your option4
```

11.

```
#include <stdio.h>
#define max 10
int que[max];
int front=-1,rear=-1;
void reverse();
void insert();
int main()
{
	int option,val;
	do{
		printf("\n\n");
		printf("***menu***\n");
		printf("1. insert an element\n");
		printf("2. display reverse queue\n");
		printf("3. exit\n");
		printf("Enter your option : ");
		scanf("%d",&option);
		switch(option)
		{
		case 1:
			insert();
			break;
		case 2:
			reverse();
			break;
		}

	}while(option!=3);

}

void insert()
{
	int num;
	printf("Enter the number to be inserted in the queue : ");
	scanf("%d",&num);
	if(rear== max-1)
		printf("overflow");
	else if(rear==-1 && front==-1)
		rear=front=0;
	else
		rear++;
```

```
        que[rear]=num;
}

void reverse()
{
        int i;
        for(i=rear;i>=0;i--)
        printf("%d",que[i]);

}
```

```
***menu***
1. insert an element
2. display reverse queue
3. exit
Enter your option : 1
Enter the number to be inserted in the queue : 1


***menu***
1. insert an element
2. display reverse queue
3. exit
Enter your option : 1
Enter the number to be inserted in the queue : 2


***menu***
1. insert an element
2. display reverse queue
3. exit
Enter your option : 1
Enter the number to be inserted in the queue : 3


***menu***
1. insert an element
2. display reverse queue
3. exit
Enter your option : 2
321

***menu***
1. insert an element
2. display reverse queue
3. exit
Enter your option : 3
계속하려면 아무 키나 누르십시오 . . .
```

12.

```
#include <stdio.h>
#define max 10
int queA[max];
int queB[max];
int frontA=-1,rearA=-1;
int frontB=-1, rearB=-1;
void insert_A();
void insert_B();
void compare();
int main()
{
        int option,val;
        do{
                printf("\n\n");
                printf("***menu***\n");
                printf("1. insert an element in a que\n");
                printf("2. insert an element in b que\n");
                printf("3. compare a que and b que\n");
                printf("4. exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                        insert_A();
                        break;
                case 2:
                        insert_B();
                        break;
                case 3:
                        compare();
                        break;
                }

        }while(option!=4);

}

void insert_A()
{
        int num;
        printf("Enter the number to be inserted in the queue : ");
```

```
        scanf("%d",&num);
        if(rearA== max-1)
                printf("overflow");
        else if(rearA==-1 && frontA==-1)
                rearA=frontA=0;
        else
                rearA++;
        queA[rearA]=num;
}

void insert_B()
{
        int num;
        printf("Enter the number to be inserted in the queue : ");
        scanf("%d",&num);
        if(rearB== max-1)
                printf("overflow");
        else if(rearB==-1 && frontB==-1)
                rearB=frontB=0;
        else
                rearB++;
        queB[rearB]=num;
}

void compare()
{
        int i;
        int flag=1;
        for(i=frontA;i<=rearA;i++)
                {
                        if(queA[i]!=queB[i])
                                flag=0;
                }

        if(flag==1)
                printf("same");
        else
                printf("not same");
}
```

```
***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 1
Enter the number to be inserted in the queue : 1


***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 1
Enter the number to be inserted in the queue : 2


***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 1
Enter the number to be inserted in the queue : 3


***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 2
Enter the number to be inserted in the queue : 4


***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 2
Enter the number to be inserted in the queue : 1
```

```
***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 2
Enter the number to be inserted in the queue : 2


***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 3
not same

***menu***
1. insert an element in a que
2. insert an element in b que
3. compare a que and b que
4. exit
Enter your option : 4
계속하려면 아무 키나 누르십시오 . . .
```