

```

665 //Queues
666 #include<stdio.h>
667 #include<stdlib.h>
668 #define QUE_SIZE 3
669 int item, frontofque=0, rear=-1, q[10];
670 void insertrear() {
671     if(rear==QUE_SIZE-1) {
672         printf("Queue Overflow\n");
673         return;
674     }
675     rear=rear+1;
676     q[rear]=item;
677 }
678 int deletefront() {
679     if(frontofque>rear) {
680         frontofque=0;
681         rear=-1;
682         return -1;
683     }
684     return q[frontofque++];
685 }
686 void displayQ() {
687     int i;
688     if(frontofque>rear) {
689         printf("Queue is empty\n");
690         return;
691     }
692     printf("Contents of queue :\n");
693     for(i=frontofque; i<=rear; i++) {
694         printf("%d\n", q[i]);
695     }
696 }
697 int main()
698 {

```

```
699 int choice;
700 for(;;){
701     printf("\n1:Insert Item\n2:Delete Item\n3:Display\n4:exit\n");
702     printf("Enter the choice\n");
703     scanf("%d",&choice);
704     switch(choice){
705         case 1:printf("Enter the item to be inserted\n");
706                 scanf("%d",&item);
707                 insertrear();
708                 break;
709         case 2:item=deletefront();
710                 if(item==-1){
711                     printf("Queue underflow\n");
712                     break;
713                 }
714                 else{
715                     printf("Item deleted =%d\n", item);
716                     break;
717                 }
718         case 3:displayQ();
719                 break;
720         case 4:exit(0);
721         default: printf("Enter valid instruction!!!");
722     }
723 }
724 return 0;
725 }
```

```
1:Insert Item
2:Delete Item
3:Display
4:exit
Enter the choice
1
Enter the item to be inserted
23
```

```
1:Insert Item
2:Delete Item
3:Display
4:exit
Enter the choice
1
Enter the item to be inserted
46
```

```
1:Insert Item
2:Delete Item
3:Display
4:exit
Enter the choice
3
Contents of queue :
23
46
```

```
1:Insert Item
2:Delete Item
3:Display
4:exit
Enter the choice
2
Item deleted =23
```

```
1:Insert Item
2:Delete Item
3:Display
4:exit
Enter the choice
2
Item deleted =46
```

```
1:Insert Item
2:Delete Item
3:Display
4:exit
Enter the choice
4
```

```
Process returned 0 (0x0)    execution time : 9.174 s
Press any key to continue.
```

```

727 //Circular Queue
728 #include<stdio.h>
729 #include<stdlib.h>
730 #define q_s 5
731 int item,f=0,r=-1,q[q_s],c=0;
732 void insert()
733 {
734     if(c==q_s){
735         printf("Queue overflow\n");
736         return;
737     }
738     r=(r+1)%q_s;
739     q[r]=item;
740     c++;
741 }
742 int delete_front()
743 {
744     if(c==0){
745         return -1;
746     }
747     item=q[f];
748     f=(f+1)%q_s;
749     c=c-1;
750     return item;
751 }
752 void display()
753 {
754     int i,front;
755     if(c==0){
756         printf("Queue is empty\n");
757         return;
758     }
759     front=f;
760     printf("Contents of queue : \n");

```



```

761     for(i=0;i<c;i++){
762         printf("%d\n",q[front]);
763         front=(front+1)%q_s;
764     }
765 }
766 int main()
767 {
768     int ch;
769     for(;;){
770         printf("\n1.insert_rear\n2.delete_front\n3.display\n4.exit\n");
771         printf("Enter the choice : ");
772         scanf("%d",&ch);
773         switch(ch){
774             case 1:printf("Enter the item : ");
775                     scanf("%d",&item);
776                     insert();
777                     break;
778             case 2:item=delete_front();
779                     if(item==-1){
780                         printf("Queue is empty\n");
781                     }
782                     else{
783                         printf("Item deleted : %d\n",item);
784                     }
785                     break;
786             case 3:display();
787                     break;
788             case 4:exit(0);
789             default:printf("Enter proper choice!!!\n");
790                     break;
791         }
792     }
793     return 0;
794 }

```

1.insert_rear

2.delete_front

3.display

4.exit

Enter the choice : 1

Enter the item : 23

1.insert_rear

2.delete_front

3.display

4.exit

Enter the choice : 1

Enter the item : 53

1.insert_rear

2.delete_front

3.display

4.exit

Enter the choice : 2

Item deleted : 23

1.insert_rear

2.delete_front

3.display

4.exit

Enter the choice : 2

Item deleted : 53

1.insert_rear

2.delete_front

3.display

4.exit

Enter the choice : 2

Queue is empty

1.insert_rear

2.delete_front

3.display

4.exit

Enter the choice : 4

Process returned 0 (0x0) execution time : 18.791 s

Press any key to continue.

```

797 //Dequeue
798 #include<stdio.h>
799 #include<stdlib.h>
800 #define qsize 5
801 int f=0,r=-1,ch;
802 int item,q[10];
803 int isfull()
804 {
805     return(r==qsize-1)?1:0;
806 }
807 int isempty()
808 {
809     return(f>r)?1:0;
810 }
811 void insert_rear()
812 {
813     if(isfull()){
814         printf("Queue overflow\n");
815         return;
816     }
817     r=r+1;
818     q[r]=item;
819 }
820 void delete_front()
821 {
822     if(isempty()){
823         printf("Queue underflow\n");
824         return;
825     }
826     printf("Item deleted is %d\n",q[(f)++]);
827     if(f>r){
828         f=0;
829         r=-1;
830     }

```

```
831 }
832 void insert_front()
833 {
834     if(f!=0){
835         f=f-1;
836         q[f]=item;
837         return;
838     }
839     else if((f==0)&&(r==-1)){
840         q[++(r)]=item;
841         return;
842     }
843     else
844         printf("Insertion is not possible\n");
845 }
846 void delete_rear(){
847     if(isempty()){
848         printf("Queue underflow\n");
849         return;
850     }
851     printf("Item deleted is %d\n",q[(r)--]);
852     if(f>r){
853         f=0;
854         r=-1;
855     }
856 }
857 void display(){
858     int i;
859     if(isempty()){
860         printf("Queue is empty\n");
861         return;
862     }
863     printf("Contents of queue are : \n");
864     for(i=f;i<=r;i++){
```



```
865     printf("%d\n",q[i]);
866 }
867 }
868 int main()
869 {
870     for(;;){
871         printf("\n1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
872         printf("Enter the choice : ");
873         scanf("%d",&ch);
874         switch(ch){
875             case 1:printf("Enter the item : ");
876                     scanf("%d",&item);
877                     insert_rear();
878                     break;
879             case 2:printf("Enter the item : ");
880                     scanf("%d",&item);
881                     insert_front();
882                     break;
883             case 3:delete_rear();
884                     break;
885             case 4:delete_front();
886                     break;
887             case 5:display();
888                     break;
889             case 6:exit(0);
890             default:printf("Enter proper choice!!!");
891                     break;
892         }
893     }
894     return 0;
895 }
```

```
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit
```

Enter the choice : 1

Enter the item : 23

```
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit
```

Enter the choice : 1

Enter the item : 45

```
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit
```

Enter the choice : 5

Contents of queue are :

23

45

```
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit
```

Enter the choice : 4

Item deleted is 23

```
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit
```

Enter the choice : 6

Process returned 0 (0x0) execution time : 43.076 s

Press any key to continue.