

Input restrict

```
1.insert_rear
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
20
1.insert_rear
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
10
20
30
1.insert_rear
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 30
1.insert_rear
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```

Output res

```
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
2
enter the item
20
insertion not possible
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
1
enter the item
20
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
5
10
20
30
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
```

```
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
5
20
30
1.insert_rear
2.insert_front
4.delete_front
5.display
6.exit
enter choice
```

// multiple priority queue

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#define N 3

int queue[N][N];

int front[N] = {0, 0, 0};

int rear[N] = {-1, -1, -1};

int item, i, pr;

```
void pginvert (int pr) {  
    if (rear[pr] == N-1)  
        printf("in queue overflow\n");  
    else {  
        printf("enter the item:\n");  
        scanf("%d", &item);  
        rear[pr]++;  
        queue[pr][rear[pr]] = item;  
    }  
    return;  
}
```

```
void pdelete() {  
    int i;  
    for (i=0; i<3; i++) {  
        if (rear[i] == front[i]-1) {  
            printf("queue empty\n");  
        }  
    }  
}
```

```

else {
    printf("deleted item %d of queue\n",
           queue[i][front[i]], i+1);
    front[i]++;
    return;
}
}

void display()
{
    int i, j;
    for (i=0; i<3; i++) {
        if (rear[i] == front[i]-1)
            printf("queue empty\n", i+1);
        else {
            printf("in queue %d: ", i+1);
            for (j=front[i]; j<=rear[i]; j++)
                printf("%d\t", queue[i][j]);
        }
    }
    return;
}

```

```

int main() {
    int ch;
    while(1) {
        printf("1. PQ insert\n");
        printf("2. PQ delete\n");
        printf("3. PQ display\n");
        printf("4. Exit\n");
        printf("Enter one choice\n");
        scanf("%d", &ch);
        switch(ch) {
            case 1: printf("Enter the priority\n");
                    scanf("%d", &pr);
                    if (pr > 0 & pr < 4)
                        pqinsert(pr-1);
                    else {
                        printf("only 3 priority exist\n");
                    }
                    break;
            case 2: pqdelete();
                    break;
            case 3: pqdisplay();
                    break;
            case 4: exit(0);
        }
        getch();
    }
}

```



```
PRIORITY QUEUE
*****

        1:PQinsert
        2:PQdelete
        3:PQdisplay
        4:Exit

enter the choice
1

enter the priority number
1

enter the item
45
PRIORITY QUEUE
*****

        1:PQinsert
        2:PQdelete
        3:PQdisplay
        4:Exit

enter the choice
1

enter the priority number
1

enter the item
67
PRIORITY QUEUE
*****

        1:PQinsert
        2:PQdelete
        3:PQdisplay
        4:Exit

enter the choice
1

enter the priority number
1

enter the item
78
```

```
PRIORITY QUEUE
*****

    1:PQinsert
    2:PQdelete
    3:PQdisplay
    4:Exit

enter the choice
1

enter the priority number
2

enter the item
67
PRIORITY QUEUE
*****

    1:PQinsert
    2:PQdelete
    3:PQdisplay
    4:Exit

enter the choice
1

enter the priority number
3

enter the item
78
PRIORITY QUEUE
*****

    1:PQinsert
    2:PQdelete
    3:PQdisplay
    4:Exit

enter the choice
1

enter the priority number
2

enter the item
4
```



```

PRIORITY QUEUE
*****

    1:PQinsert
    2:PQdelete
    3:PQdisplay
    4:Exit

enter the choice
3

QUEUE 1:45      67      78
QUEUE 2:67      4
QUEUE 3:78      PRIORITY QUEUE
*****

    1:PQinsert
    2:PQdelete
    3:PQdisplay
    4:Exit

enter the choice
2
deleted item is 45 of queue 1
PRIORITY QUEUE
*****

    1:PQinsert
    2:PQdelete
    3:PQdisplay
    4:Exit

enter the choice
2
deleted item is 67 of queue 1
PRIORITY QUEUE
*****

    1:PQinsert
    2:PQdelete
    3:PQdisplay
    4:Exit

enter the choice
3

QUEUE 1:78
QUEUE 2:67      4
QUEUE 3:78      PRIORITY QUEUE
*****

```

// Asc & Desc Priority Que

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 3

int pq[MAX];
int count = 0; // add int d=0 per desc

void insert(int data){
    int i = 0;
    if (count == MAX){
        printf("queue overflow\n");
        return;
    }
    if (count == 0){
        pq[count++] = data;
    }
    else {
        for (i = count - 1; i >= 0; i--){
            if (data > pq[i]){
                pq[i+1] = pq[i];
            }
            else {
                break;
            }
        }
        pq[i+1] = data;
        count++;
    }
}
```

```

int removeData() {
    return pq[--count]; // pq[d++]
    // for disc
}

```

```

void display() {
    int i;
    if (count == 0) {
        printf("queue is empty\n");
        return;
    }
}

```

```

printf("contents of queue: ");
for (i = count - 1; i >= 0; i--) {
    printf("%d ", pq[i]);
}

```

```

printf("\n"); // i = d; i = count; i++
// for disc
}

```

```

int main() {
    int choice, item;
    for(;;) {

```

```

        printf("\n 1: insert 2: delete smallest\n");
        printf("3: display 4: exit\n");

```

```

        printf("Enter the choice: ");
        scanf("%d", &choice);
    }
}

```

Switch (choice) {

case 1: printf ("Enter the item to be inserted : ");

scanf ("%d", & item);

insert (item);

break;

case 2: item = remove Data();

if (item == -1)

printf ("Queue is empty \n");

else

printf ("item deleted = %d \n", item);

break;

case 3: display();

break;

default : exit (0);

}

}

}

AScending

```
1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :10

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :20

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :30

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :3
Contents of queue: 10 20 30

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :2
item deleted=10

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :3
Contents of queue: 20 30

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :
```

Descending

```
1:insert 2:delete_largest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :10

1:insert 2:delete_largest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :20

1:insert 2:delete_largest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :30

1:insert 2:delete_largest 3:display 4:exit
Enter the choice :3
Contents of queue: 30 20 10

1:insert 2:delete_largest 3:display 4:exit
Enter the choice :2
item deleted=30

1:insert 2:delete_largest 3:display 4:exit
Enter the choice :2
item deleted=20

1:insert 2:delete_largest 3:display 4:exit
Enter the choice :3
Contents of queue: 10

1:insert 2:delete_largest 3:display 4:exit
Enter the choice :
```