

→ Ascending Priority Queue -

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 4
int pq[MAX];
int count = 0;
int d = 0;
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[d++];
}
```

```

void display() {
    int i;
    if (count == 0) {
        printf ("Queue is empty \n");
        return;
    }
    printf ("Content of queue \n");
    for (i = 0; i < count; i++)
        printf ("%d", pq[i]);
    printf ("\n");
}

int main () {
    int choice, item;
    for (;;) {
        printf ("1.insert 2.delete_smallest \n"
               "3.display \n 4.exit \n");
        printf ("Enter choice ");
        scanf ("%d", &choice);
        switch (choice) {
            case 1: printf ("Enter item \n");
                      scanf ("%d", &item);
                      insert (item);
                      break;
            case 2: item = removeData ();
                      if (item == -1)
                          printf ("Queue is empty ");
                      else
                          printf ("item deleted = %d \n", item);
                      break;
            case 3: display ();
                      break;
            default : exit (0);
        }
    }
}

```

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 4

int pq[MAX];
int count = 0;
int d = 0;

void insert(int data){
    int i = 0;
    if(count==MAX)
    {
        printf("Queue overflow\n");
        return;
    }
    // if queue is empty, insert the data
    if(count == 0){
        pq[count++] = data;
    }else{
        // start from the right end of the queue
        for(i = count - 1; i >= 0; i-- ){
            //if data is smaller shift right
            if(data<pq[i]){
                pq[i+1] = pq[i];
            }else{
                break;
            }
        }
        // insert the data
        pq[i+1] = data;
        count++;
    }
}

int removeData(){
    return pq[d++];
}

void display()
{int i;
if (count==0)
```

```
void display()
{
    int i;
    if (count==0)
    {
        printf("queue is empty\n");
        return;
    }
    printf("Contents of queue: ");
    for(i=d;i<count;i++)
    {
        printf("%d ",pq[i]);
    }
    printf("\n");
}

int main() {
    int choice,item;
    for(;;)
    {
        printf("\n1:insert 2:delete_smallest 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=removeData();
            if(item==-1)
            printf("Queue is empty\n");
            else
            printf("item deleted=%d\n",item);
            break;
            case 3:display();
            break;
            default:exit (0);
        }
    }
}
```

```
1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :5

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :7

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :6

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :8

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :9
Queue overflow

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :3
Contents of queue: 5 6 7 8

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :2
item deleted=5

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :3
Contents of queue: 6 7 8

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :2
item deleted=6

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :2
item deleted=7

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :2
item deleted=8

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