Ascending Priority Queue

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 4
int pq[MAX];
int count = 0;
int front = 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]) {
        pq [i+1] = pq [i];
      }
      else
          break;
}
```

```
        pq[i+1] = data;
        count++;
    }
}
int removeData() {
    if (count == 0)
        return -1;
    if (front >= count)
        return -1;
    return pq[front++];
}
void display() {
    int i;
    if (count == 0)
    {
        printf("Queue is empty \n");
        return;
    }
    printf("Contents of queue : ");
    for (i = front; i < count; i++)
    {
        printf("%d", pq[i]);
    }
    printf("\n");
}
```

```
int main () {
  int choice, item;
for(;;)
{
  printf ("\n 1: 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);
  }
}
}
```

Output:

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

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: 2.

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: 3
Contents of the queue: 2 5 10.

1: insert 2: delete_smallest 3: display 4: exit
item deleted = 2.