Ascending priority Queue

```
#include <stdio.h>
# include <stdlib.h>
# define q_size 5
int r=-1, f=0, item , count =0;
 int q[10], ch ;
void insert _rear ()
{
     if (r== q_size -1)
{
     printf ("Queue overflow \n");
     return;
  }
  r=r+1;
  q[r] = item;
  count ++ ;
}
  void   insertion_sort ()
{
    int i, j, key ;
for (i=1, i<count ; i++)
{
    key = q[i];
    j= i-1;
    while   (j>=0 && q[i] < key)
    {
```

```
    q[j+1]  = key;
}
void  delete_rear( )
{
  if  (f>r)
  {
     f=0;
     r=-1;
   printf ("Queue is empty \n");
     return ;
  }
 printf ("Item deleted := %d \n", q[r--]);
}
void display ()
{
   if (f>r)
  {
   printf ("Queue is empty)\n");
   return;
 }
 printf ("Contents of the queue are: \n");
 for (int i=f; i<=r; i++)
{
   printf ("%d\n", q[i]);
 }
}
```

```
int main ()
{
   for (;;)
{
    printf ("\n1: insert_rear\n 2 : delete rear :\n3.
          display \n");
printf ("Enter the choice : \n");
scanf ("%d", &ch);
switch (ch) {
      case 1: - printf ("Enter the item: \n");
      scanf ("%d", & item);
      insert rear();
      insertion_sort ();
      break;
case 2:-    delete rear ();
      break;
case 3:-  display ();
      break;
    default : exit (0);
   }
  }
return 0;
}
```