```
//priority Queue

#include<stdio.h>
#define MAX 5

struct queue
{
    int data[MAX];
    int front, rear;
};
typedef struct queue queue;

int full(queue q)
{
      if(q.rear==MAX-1)
            return(1);
      else
            return(0);
}

int empty(queue q)
{
      if(q.front==-1)
            return(1);
      else
            return(0);
}


void pqinsert (queue q[], int num, int p)
{
   int i=p-1;

   if(full(q[i]))
   {
      printf("\nQueue %d is full", i+1);
      return;
   }
   q[i].data[++q[i].rear]=num;

   if(q[i].rear==0)
          q[i].front=0;
}

int pqdelete(queue q[])
{
   int i num,

   for(i=0; empty(q[i]) && i<3; i++);

   if(i==3)
   {
       printf("\nPriority queue is empty");
       return(-1);
```

```
   }

   num=q[i].data[q[i].front];

   if(q[i].rear==q[i].front)
         q[i].front=q[i].rear=-1;
   else
         q[i].front++;
   return(num);
}

void pqdisplay(queue q[])
{
   int i,j;

   for(i=0; i<3; i++)
   {
       if(empty(q[i]))
       {
          printf("\nqueue %d is empty",i+1);
       }
       else
       {
          printf("\nQueue %d=",i+1);
          for(j=q[i].front; j<=q[i].rear; j++)
          {
               printf("%5d",q[i].data[j]);
         }
       }
    }
}

main()
{
  queue q[3];
  int ch,p,done=1,i,num;

  for(i=0; i<3; i++)
      q[i].front=q[i].rear=-1;

  while(done)
  {
     printf("\n1:PQINSERT \n2:PQDELETE \n3:PQDISPLAY \n4:EXIT");
     printf("\nEnter your choice:");
     scanf("%d",&ch);

     switch(ch)
     {
        case 1: printf("\nEnter the number and its priority");
                scanf("%d%d",&num,&p);
                pqinsert(q,num,p);
                break;

        case 2: num=pqdelete(q);
```

```
                if(num!=-1)
                {
                    printf("\n%d is deleted",num);
                }
                break;

       case 3: pqdisplay(q);
               break;

       case 4:
       default: done=0;
     }
 }
}
```