



Multiple ~~priority~~ priority queue

```
#include <stdio.h>
#include <stdlib.h>
#define N 3
int queue [3][N];
int front [3] = {0,0,0};
int rear [3] = {-1,-1,-1};
void pqinsert (int pr);
void pqdelete ();
void display ();
int item, pr;
int main ()
{
    int ch;
    while (1)
    {
        printf ("PRIORITY QUEUE \n");
        printf ("*** \n");
        printf ("1: Pq insert \n");
        printf ("2: Pq delete \n");
        printf ("3: Pq display \n");
        printf ("4: Exit \n");
        printf ("Enter the choice \n");
        scanf ("%d", &ch);
        switch (ch)
        {
            case 1: printf ("Enter the priority number \n");
                    scanf ("%d", &pr);
                    if (pr > 0 && pr < 4)
                        pqinsert (pr-1);
```

else
printf ("n 3 priority exists 1 2 3\n");
break;

Case 2: pdelete ()
delete;

Case 3: display ()
break;

Case 4: exit (0);

}

}

}

void pinsert (int pr)
{

if (rear [pr] == N - 1)

printf ("n Queue Overflow\n");
else

{

printf ("n 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 ("In queue empty\n");
else
{
  printf ("deleted item is %d of queue %d\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 ("In queue empty %d\n", i+1);
    else
    {
      printf ("In Queue %d:", i+1);
      for (j=front [i]; j<=rear [i]; j++)
        printf ("%d ", queue [i][j]);
    }
  }
  return;
}

```