

Multiple priority Queue

```
# include <stdio.h>
```

```
# define N 3
```

```
int queue [3][N];
```

```
int front [3] = {0, 0, 0};
```

```
int rear [3] = {-1, -1, -1};
```

```
int item, pr;
```

```
void main()
```

```
{
```

```
int ch;
```

```
while (1)
```

```
{
```

```
printf (" @ Priority Queue\n");
```

```
printf (" * * * * *\n");
```

```
printf ("In 1: PQ insert\n");
```

```
printf ("In 2: PQ delete\n");
```

```
printf ("In 3: PQ display\n");
```

```
printf ("In 4: exit\n");
```

```
printf ("In enter the choice\n");
```

```
scanf ("%d", &ch);
```

```
switch (ch)
```

```
{
```

```
case 1: printf ("In enter the priority number\n");
```

```
scanf ("%d", &pr);
```

```
if (pr > 0 && pr < 4)
```

```
    pq insert (pr - 1);
```

```
else
```

```
    printf ("I only 3 priority exist 1 2 3\n");  
    break;
```

```
case 2 : pq delete ();  
        break;
```

```
case 3 : display ();  
        break;
```

```
case 4 : exit (0);
```

```
}
```

```
}
```

```
}
```

```
pq insert (int pr)
```

```
{
```

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

```
        printf ("In queue overflow\n");
```

```
    else
```

```
    {
```

```
        printf ("enter the item\n");
```

```
        scanf ("%d", &item);
```

```
        rear[pr]++;
```

```
        queue[pr][rear[pr]] = item;
```

```
    }
```



```
return;
```

```
}
```

```
pq delete()
```

```
{
```

```
int i;
```

```
for (i = 0; i < 3; i++)
```

```
{
```

```
if (rear[i] == front[i] - 1)
```

```
printf("queue empty\n");
```

```
else
```

```
{
```

```
printf("deleted item is %d of queue %d\n",
```

```
queue[i][front[i]], i+1);
```

```
front[i]++;
```

```
return;
```

```
}
```

```
}
```

```
}
```

```
display()
```

```
{
```

```
int i, j;
```

```
for (i = 0; i < 3; i++)
```

```
{
```

```
if (rear[i] == front[i] - 1)
```

```
printf("queue empty %d\n", i+1);  
else
```

```
{
```

```
printf("In Queue %d:", i+1);
```

```
for (j = front[i]; j <= rear[i]; j++)
```

```
printf("%d\t", queue[i][j]);
```

```
}
```

```
}
```

```
return;
```

```
}
```


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 p;
```

```
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 (1 {
    int choice, item;
    for (; )
    {
        printf ("1. insert 2. delete smallest 3. display  
4. exit\n");
        printf ("Enter choice");
        scanf ("%d", &choice);
        switch (choice)
        {
            case 1: printf ("Enter the item to be inserted:");
                    scanf ("%d", &item);
                    insert (item);
                    break;
            case 2: item = remove Data();
                    if (item == -1)
                        printf ("Queue is empty\n");
                    else,
                        printf ("Item deleted = %d\n", item);
                    break;
            case 3: display();
                    break;
            default: exit (0);
        }
    }
}

```

C:\Users\Danhan\Downloads\Files\Writelnpq.mzs

```
6
PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
3
QUEUE 1:5
QUEUE 2:6
QUEUE 3:2 PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
2
deleted item is 5 of queue 1
PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
3
queue empty 1
QUEUE 2:6
QUEUE 3:2 PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
4
```

Process returned 0 (0x0) execution time : 131.622 s
Press any key to continue.

Type here to search

C:\Users\Danhan\Downloads\FileWritepq.ms

```
PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
1
enter the item
5
PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
3
enter the item
2
PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
2
enter the item
6
PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
```

Type here to search

18:44 08-11-2020

```
C:\Users\Danhan\Downloads\FileWriter.exe
1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :1

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 :3

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

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

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

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

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :4

Process returned 0 (0x0)   execution time : 66.229 s
Press any key to continue.
```

SVA_DQA (4) - Notepad

```
File Edit Format View Help

#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++;
}

}
```

3398 chars, 166 line breaks 16:31 08:27 PM 06-11-2020


```
SVA_DQA (4) - Notepad
File Edit Format View Help

// insert the data
pq[i+1] = data;
count++;
}

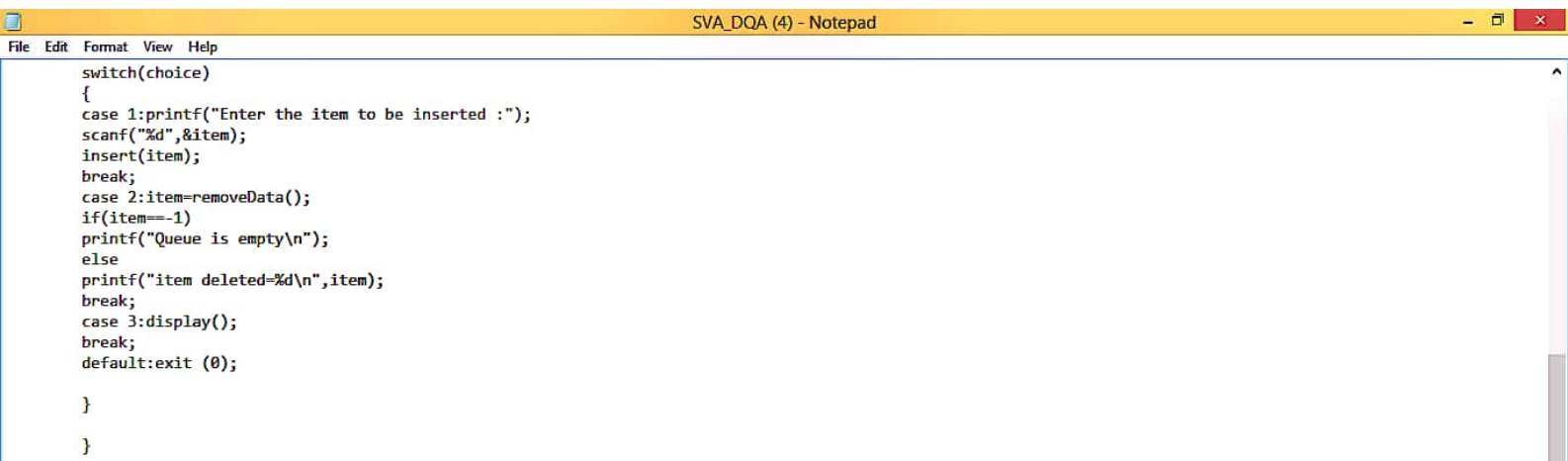
}

int removeData(){

return pq[d++];
}

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 :");
```




```
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);

}

}
```

```
SVA_MPO - Notepad
File Edit Format View Help

#include<stdio.h>
#include<conio.h>
#define N 3
int queue[3][N];
int front[3]={0,0,0};
int rear[3]={-1,-1,-1};
int item,pr;
void main()
{
    int ch;
    clrscr();
    while(1)
    {
        printf("PRIORITY QUEUE\n");
        printf("*****\n");
        printf("\n\t1:PQinsert\n");
        printf("\n\t2:PQdelete\n");
        printf("\n\t3:PQdisplay\n");
        printf("\n\t4:Exit\n");
        printf("\nenter the choice\n");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:printf("\nenter the priority number\n");
                    scanf("%d",&pr);
                    if(pr>0 && pr<4)
                        pqinsert(pr-1);
                    else
                        printf("\nonly 3 priority exists 1 2 3\n");
                    break;
            case 2:pqdelete();
                    break;
            case 3:display();
                    break;
            case 4:exit(0);
        }
    }
}
```




```
SVA_MPQ - Notepad
File Edit Format View Help
break;
case 4:exit(0);
}
}
getch();
}
pqinsert(int pr)
{
if(rear[pr]==N-1)
printf("\n Queue overflow\n");
else
{
printf("\nenter the item\n");
scanf("%d",&item);
rear[pr]++;
queue[pr][rear[pr]]=item;
}
return;
}
pqdelete()
{
int i;
for(i=0;i<3;i++)
{
if(rear[i]==front[i]-1)
printf("\nqueue empty\n");
else
{
printf("deleted item is %d of queue %d\n",queue[i][front[i]],i+1);
front[i]++;
return;
}
}
}
display()
{
<
```

SVA_MPQ - Notepad

```
File Edit Format View Help

rear[pr]++;
queue[pr][rear[pr]]=item;
}
return;
}
pqdelete()
{
int i;
for(i=0;i<3;i++)
{
if(rear[i]==front[i]-1)
printf("\nqueue empty\n");
else
{
printf("deleted item is %d of queue %d\n",queue[i][front[i]],i+1);
front[i]++;
return;
}
}
}
display()
{
int i,j;
for(i=0;i<3;i++)
{
if(rear[i]==front[i]-1)
printf("\nqueue empty %d\n",i+1);
else
{
printf("\nQUEUE %d:",i+1);
for(j=front[i];j<=rear[i];j++)
printf("%d\t",queue[i][j]);
}
}
return;
}
```

Windows taskbar icons: File Explorer, Google Chrome, Microsoft Excel, Microsoft Word, Notepad. System tray: 08:24 PM, 08-11-2020.