

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("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 ($p_r > 0$ $\vee p_r < 4$)

 pq insert ($p_r - 1$);

else

 printf ("There are only 3 priority exists [2 3] \n");
 break;

case 2 : pq delete ();

 break;

case 3 : display ();

 break;

case 4 : exit (0);

}

}

}

 pq insert (int p_r)

{

 if ($rear[p_r] == N - 1$)

 printf ("In queue overflow \n");

 else

{

 printf ("Enter the item \n");

 scanf ("%d", &item);

 rear [p_r]++;

 queue [p_r][$rear[p_r]$] = 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 %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 \cdot d \n", i+1);  
else  
{  
    printf ("In Queue l-d : ", p+1);  
    for (j = front [p]; j <= rear [i]; j++)  
        printf ("\t", queue [i] [j]);  
}  
}  
return;  
}
```

Avending Priority queue

```
#include < stdio.h >
#include < string.h >
#include < stdlib.h >
#define MAX 5
```

```
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] = 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()
{
    int choice, item;
    for(;;)
    {
        printf("1. insert 2. delete smallest 3. display\n4. 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();
                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\Dashan\Downloads\Files\Write\pq.cxx
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.
```

```
C:\Users\Dashan\Downloads\Files\Write\pq.cxx
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
```



```
C:\Users\Dashan\Downloads\Files\Wtq.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

```
#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++;
    }
}
```

IntelliJ 2020.3.1 3398 chars, 166 line breaks 16:10 03/07/2020 08-11-2020

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 :");
}
```

IntelliJ 2020.3.1 3398 chars, 166 line breaks 16:10 08-11-2020 08:22 PM

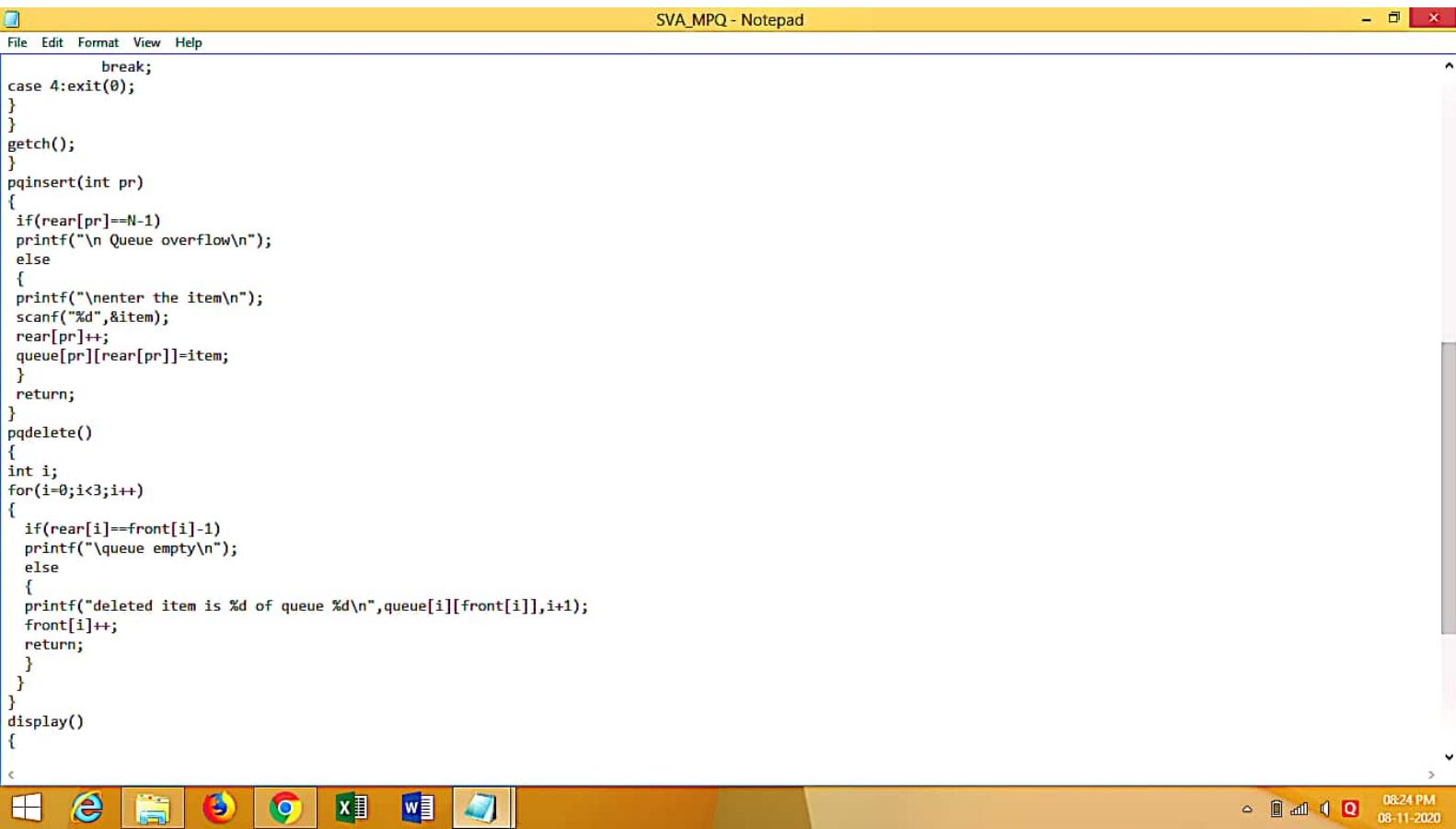
SVA_DQA (4) - Notepad

```
File Edit Format View Help
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_MPQ - 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);
        }
    }
}
```

08:24 PM
08-11-2020



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("\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("\nQUEUE %d:",i+1);
for(j=front[i];j<=rear[i];j++)
printf("%d\t",queue[i][j]);
}
}
return;
}

```

08:24 PM
08-11-2020