classmate

Date 4/11/2020

Page

1) 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 pqinsert (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 pqdelete(){
 int i;
 for(i=0; i<3; i++)
  {
  if(rear[i]==font[i]-1)
  printf("queue empty\n");
  else
  {
  printf("deleted item is %d of queue %d\n", queue[i][font[i]],
    i+1);
   front[i]++;
 return; }
  }
 }

void 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("\n QUEUE %d: ", i+1);
```

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

int main () {
 int ch;
 printf ("PRIORITY QUEUE \n");
 printf ("*********\n");
 printf ("\n\t 1: PQ insert\n");
 printf ("\n\t 2: PQ delete\n");
 printf ("\n\t 3: PQ display\n");
 printf ("\n\t 4: Exit\n");
  while (1)
 {
 printf ("\n enter the choice \n");
 scanf("%d", &ch);
  switch (ch)
  {
  case 1: printf ("\n enter the priority number \n");
  scanf("%d", &pr);
  if (pr>0 && pr<4)
```

```
pqinsert (pr-1);
else
printf ("only 3 priority exists  1 2 3 \n");
break;
case 2: pqdelete(); break;
case 3: display(); break;
case 4: break;
  }
 }
return 0;
}
```

2) ASCENDING PRIORITY QUEUE

```
#include <stdio.h>
#define MAX 4
int pq [MAX];
int count =0 , d =0 ;

void insert (int data) {
int i =0;
 if (count = =MAX )
 { printf (" Queue overflow \n");
   return;
 }

 //if queue is empty, insert data
if (count ==0) {
 pq [count ++] =data ;
 } else {
 // start from 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 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 in ascending ~~order~~ priority ");
for (i=d; i<count; i++)
  {
  printf("%d", pq [i]);
  }
 printf ("\n");
}
```

```
int main() {
int choice, item;
 do
 {
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:");
         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 : break;
  }
 } while (choice!=4);
 return 0;
 }
```

3) DESCENDING PRIORITY

```
#include <stdio.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 remove Data () {
return pq[d++];
}

void display ()
{ int i;
if (count ==0)
 {
printf ("queue is empty\n");
return;
}
printf ("Contents of queue in descending priority : ");
for (i=d ; i<count ; i++)
  {
printf ("%d", pq[i]);
}
printf ("\n");
 }


int main () {
int choice, item;
```

```
do
{
printf ("\n1. insert 2: delete_largest 3 : display  4: exit\n");
printf ("Enter the choice : ");
scanf ("%d ", &choice);
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;
~~d~~ case 3 : display() ; break;
default : break;
}
} ~~l~~while (choice! = 4);
return 0;
}
```