EXNO:8 IMPLEMENTATION OF PRIORITY QUEUE USING

HEAPS

AIM:

To write a C program to implement Priority Queue using Binary Heaps.

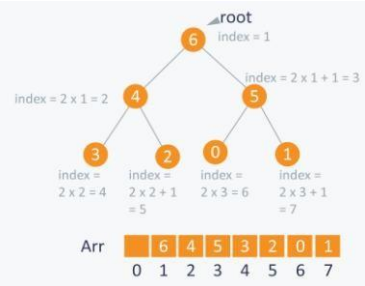
DESCRIPTION:

A heap is a tree-based data structure in which all the nodes of the tree are in a specific order.

For example, if X is the parent node of Y, then the value of X follows a specific order with respect to the value of Y and the same order will be followed across the tree.

The maximum number of children of a node in a heap depends on the type of heap. However, in the more commonly-used heap type, there are at most 2 children of a node and it's known as a Binary heap

An array can be used to simulate a tree in the following way. If we are storing one element at index i in array Arr, then its parent will be stored at index i/2 (unless its a root, as root has no parent) and can be accessed by Arr[i/2], and its left child can be accessed by Arr[2∗i] and its right child can be accessed by Arr[2∗i+1]. Index of root will be 1 in an array.



ALGORITHM:

1. Initialize all necessary variables and functions.
2. Read the choices.
3. For insertion, read the element to be inserted.
4. If root is NULL, assign the given element as root.
5. If the element is equal to the root, print “Duplicate value”.

6. Else if element value is less than the root value, insert element at the left of the root.

7. Else insert right side of the root.

8. For deletion, get the priority for maximum or minimum.

9. If maximum, it deletes the root and rearranges the tree.

10. If minimum, it deletes the leaf.

11. End of the program

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include
<stdlib.h> enum
{FALSE=0,TRUE=-1};
struct Node     {
      struct Node
*Previous;        int
Data;       struct Node
*Next;
    }Current;
struct Node *head;
struct Node *ptr;
static int NumOfNodes;
int
PriorityQueue(void);
int Maximum(void);
int Minimum(void);
void Insert(int);
int Delete(int);
void Display(void);
int Search (int);
void main()
{
    int choice;
int DT;
PriorityQueue()
;

while(1
)
                {
                printf("\nEnter ur Choice:");

      printf("\n1.Insert\n2.Display\n3.Delete\n4.Search\n5.Exit
\n");       scanf("%d",&choice);
                switch(choice)
                {
                case 1:
                  printf("\nEnter a data to enter Queue");

scanf("%d",&DT);
        Insert(DT);
```

```
    break;
case 2:
    Display();        break;
case 3:         {
int choice,DataDel;
printf("\nEnter ur choice:");
      printf("\n1.Maximum Priority queue\n2.Minimum priority Queue\n");
      scanf("%d",&choice);
                switch(choice)
                {
```

```
case 1:
               DataDel=Maximum();
Delete(DataDel);                 printf("\n%d
is deleted\n",DataDel);
                break;
            case 2:



 DataDel=Minimum();
               Delete(DataDel);
printf("\n%d is deleted\n",DataDel);
                break;
            default:
                      printf("\nSorry Not a correct Choice\n");
                      }
                    }
          break;
      case 4:
                    printf("\nEnter a data to Search in Queue:");
          scanf("%d",&DT);
if(Search(DT)!=FALSE)       printf("\n %d is
present in queue",DT);            else
          printf("\n%d is not present in
queue",DT);           break;        case 5:
          exit(0);      default:
                printf("\nCannot process ur choice\n");
                } }}
int PriorityQueue(void)
{
    Current.Previous=NULL;
    printf("\nEnter first element of Queue:");

scanf("%d",&Current.Data)
;     Current.Next=NULL;
head=&Current;
ptr=head;
NumOfNodes++;     return;
}
int Maximum(void)
{     int
Temp;
ptr=head;
Temp=ptr-
>Data;

    while(ptr->Next!=NULL)
    {       if(ptr-
>Data>Temp)
Temp=ptr->Data;
                ptr=ptr->Next;
    }
    if(ptr->Next==NULL && ptr->Data>Temp)
```

```
      Temp=ptr->Data;
return(Temp);
}
int Minimum(void)
{     int Temp;
ptr=head;
Temp=ptr->Data;
while(ptr-
>Next!=NULL)
```

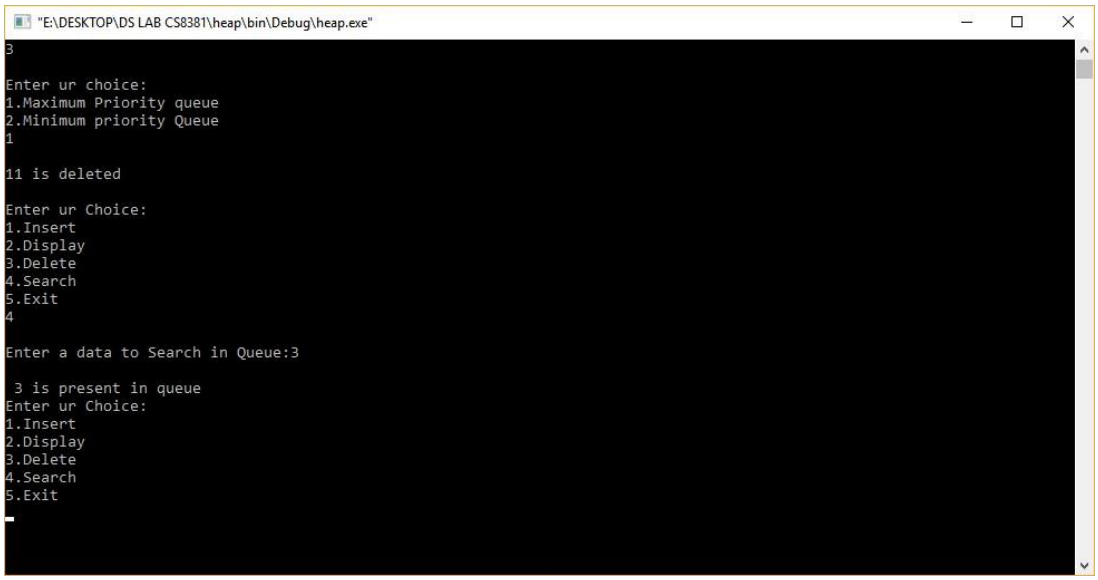
```
 {    if(ptr-
>Data<Temp)
Temp=ptr->Data;

ptr=ptr->Next;
    }
    if(ptr->Next==NULL && ptr->Data<Temp)
      Temp=ptr-
>Data;
return(Temp); }
void Insert(int DT)
{

  struct Node *newnode;
    newnode=(struct Node *)malloc(sizeof(struct
Node));     newnode->Next=NULL;     newnode-
>Data=DT;
    while(ptr->Next!=NULL)
      ptr=ptr->Next;
    if(ptr->Next==NULL)
    {
                newnode->Next=ptr->Next;
                ptr->Next=newnode;
    }
    NumOfNodes++;
}
int Delete(int DataDel)
{
    struct Node *mynode,*temp;
ptr=head;     if(ptr-
>Data==DataDel)
    {
                temp=ptr;
                ptr=ptr->Next;
       ptr->Previous=NULL;
      head=ptr;
      NumOfNodes--;
      return(TRUE);
    }
else
{
                while(ptr->Next->Next!=NULL)
                {
                   if(ptr->Next->Data==DataDel)
          {
      mynode=ptr;
                      temp=ptr->Next;
                      mynode->Next=mynode->Next->Next;
            mynode->Next-
>Previous=ptr;
      free(temp);
```

```
      NumOfNodes--;
      return(TRUE);
                    }
                    ptr=ptr->Next;
                }
                if(ptr->Next->Next==NULL && ptr->Next->Data==DataDel)
                {
          temp=ptr->Next;
free(temp);           ptr-
>Next=NULL;
NumOfNodes--;
```

```
return(TRUE);
                 }
    }
return(FA
LSE);
}
int Search(int DataSearch)
{
ptr=head
;
    while(ptr->Next!=NULL)
    {

 if(ptr->Data==DataSearch)
                    return ptr->Data;
                ptr=ptr->Next;
    }
     if(ptr->Next==NULL && ptr-
>Data==DataSearch)      return ptr->Data;
    return(FALSE);
}
void Display(void)
{
ptr=head
;
    printf("\nPriority Queue is as Follows:-\n");
while(ptr!=NULL)
    {
      printf("\t\t%d",ptr->Data);
      ptr=ptr->Next;
    }
}
```

OUTPUT

```
"E:\DESKTOP\DS LAB CS8381\heap\bin\Debug\heap.exe"

Enter first element of Queue:3

Enter ur Choice:
1.Insert
2.Display
3.Delete
4.Search
5.Exit
1

Enter a data to enter Queue  11

Enter ur Choice:
1.Insert
2.Display
3.Delete
4.Search
5.Exit
2

Priority Queue is as Follows:-
                3               11
Enter ur Choice:
1.Insert
2.Display
3.Delete
4.Search
5.Exit
3
```


"E:\DESKTOP\DS LAB CS8381\heap\bin\Debug\heap.exe"
3

Enter ur choice:
1.Maximum Priority queue
2.Minimum priority Queue
1

11 is deleted

Enter ur Choice:
1.Insert
2.Display
3.Delete
4.Search
5.Exit
4

Enter a data to Search in Queue:3

3 is present in queue
Enter ur Choice:
1.Insert
2.Display
3.Delete
4.Search
5.Exit

RESULT:

Thus the Priority Queue using Binary Heap is implemented and the result is verified successfully.