#a program to implement priority queue using Array.

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
int priorityQueue[SIZE];
int rear = -1;
int isFull() {
  return (rear == SIZE - 1);
}
int isEmpty() {
  return (rear == -1);
}
void insert(int element) {
  if (isFull()) {
     printf("\nPriority queue is full. Cannot insert %d.\n", element);
  } else {
     int i;
     if (isEmpty()) {
       rear = 0;
       priorityQueue[rear] = element;
     } else {
       for (i = rear; i >= 0; i--) {
          if (element < priorityQueue[i]) {</pre>
            priorityQueue[i + 1] = priorityQueue[i];
```

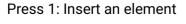


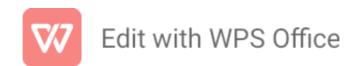
```
} else {
            break;
         }
       }
       priorityQueue[i + 1] = element;
       rear++;
     }
     printf("%d inserted into the priority queue.\n", element);
  }
}
int deleteHighestPriority() {
  if (isEmpty()) {
     printf("\nPriority queue is empty. Cannot delete.\n");
     return -1;
  } else {
     int deleted = priorityQueue[0];
    // Shift all elements left by one position
    for (int i = 0; i < rear; i++) {
       priorityQueue[i] = priorityQueue[i + 1];
     }
     rear--;
     printf("\nThe element with the highest priority (%d) is deleted\n",
deleted);
     return deleted;
  }
}
```

```
void display() {
  if (isEmpty()) {
     printf("\nPriority queue is empty.\n");
  } else {
     printf("\nElements in the priority queue are: ");
     for (int i = 0; i \le rear; i++) {
       printf("%d ", priorityQueue[i]);
     }
     printf("\n");
  }
}
int main() {
  int choice = 1, x;
  while (choice < 4 && choice != 0) {
     printf("\nPress 1: Insert an element");
     printf("\nPress 2: Delete the element with the highest priority");
     printf("\nPress 3: Display the elements");
     printf("\nEnter your choice: ");
     scanf("%d", &choice);
     switch (choice) {
       case 1:
         printf("Enter the element to be inserted: ");
         scanf("%d", &x);
         insert(x);
```



```
break;
        case 2:
           deleteHighestPriority();
           break:
        case 3:
           display();
           break;
        default:
           printf("\nInvalid choice\n");
     }
  }
   return 0;
}
o/p=
Press 1: Insert an element
Press 2: Delete the element with the highest priority
Press 3: Display the elements
Enter your choice: 1
Enter the element to be inserted: 12
12 inserted into the priority queue.
Press 1: Insert an element
Press 2: Delete the element with the highest priority
Press 3: Display the elements
Enter your choice: 1
Enter the element to be inserted: 13
13 inserted into the priority queue.
```





Press 2: Delete the element with the highest priority

Press 3: Display the elements

Enter your choice: 1

Enter the element to be inserted: 14

14 inserted into the priority queue.

Press 1: Insert an element

Press 2: Delete the element with the highest priority

Press 3: Display the elements

Enter your choice: 3

Elements in the priority queue are: 12 13 14

Press 1: Insert an element

Press 2: Delete the element with the highest priority

Press 3: Display the elements

Enter your choice: 2

The element with the highest priority (12) is deleted

Press 1: Insert an element

Press 2: Delete the element with the highest priority

Press 3: Display the elements

Enter your choice: 2

The element with the highest priority (13) is deleted

Press 1: Insert an element

Press 2: Delete the element with the highest priority

Press 3: Display the elements

Enter your choice:

