

OPERATING SYSTEM LABORATORY

Program-3

Aim: Implementation of Priority Queue

Source Code:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#define MAX 6

int intArray[MAX];
int itemCount = 0;

int peek(){
    return intArray[itemCount - 1];
}

bool isEmpty(){
    return itemCount == 0;
}

bool isFull(){
    return itemCount == MAX;
}

int size(){
    return itemCount;
}

void insert(int data){
    int i = 0;

    if(!isFull()){
        // if queue is empty, insert the data
```

```

if(itemCount == 0){
    intArray[itemCount++] = data;
}else{
    // start from the right end of the queue

    for(i = itemCount - 1; i >= 0; i-- ){
        // if data is larger, shift existing item to right end
        if(data > intArray[i]){
            intArray[i+1] = intArray[i];
        }else{
            break;
        }
    }

    // insert the data
    intArray[i+1] = data;
    itemCount++;
}
}

int removeData(){
    return intArray[--itemCount];
}

int main() {

    insert(3);
    insert(5);
    insert(9);
    insert(1);
    insert(12);
    insert(15);

    if(isFull()){
        printf("Queue is full!\n");
    }
}

```

```

// remove one item
int num = removeData();
printf("Element removed: %d\n",num);
insert(16);
insert(17);
insert(18);

printf("Element at front: %d\n",peek());

printf("-----\n");
printf("index : 5 4 3 2 1 0\n");
printf("-----\n");
printf("Queue: ");

while(!isEmpty()){
    int n = removeData();
    printf("%d ",n);
}
}

```

Output:

```

Queue is full!
Element removed: 1
Element at front: 3
-----
index : 5 4 3 2 1 0
-----
Queue:  3 5 9 12 15 16
-----
Process exited after 1.706 seconds with return value 0
Press any key to continue . . .

```