This is the handout for Online Lab 2. Submit your solutions online. You may use a list-based implementation for the priority queues.

Question 1. Max-Priority Queue

Implement a max-priority queue in Python. Your implementation should support the following operations:

- `insert(val)`: Insert a value into the priority queue.
- `pop()`: Remove and return the maximum value from the priority queue.

You may use a list-based implementation. For example, you can store the elements in a Python list and search for the maximum when dequeuing.

For example:

```
pq = MaxHeap()
pq.insert(5)
pq.insert(3)
pq.insert(8)
pq.insert(1)
print(pq.pop())  # 8
print(pq.pop())  # 5
pq.insert(10)
print(pq.pop())  # 10
```

Question 2. C Min-Priority Queue

Now, implement a min-priority queue in C. It is the same as the one you implemented in Python, but will maintain the minimum element at the top of the queue. The key value stored will be restricted to non-negative integers.

Your implementation should support the following operations:

- `createHeap()`: Create and return a new heap with a default capacity of 3.
- `insert(h, key)`: Insert a non-negative integer key into the priority queue. If the queue is full, double the capacity using `realloc`.
- `pop(h)`: Remove and return the minimum value from the priority queue. If the queue is empty, return -1.

Here is the starter code:

For example:

```
MinHeap *h = createHeap();
insert(h, 5);
insert(h, 3);
insert(h, 8);
insert(h, 1);
printf("%d\n", pop(h));  // 1
printf("%d\n", pop(h));  // 3
insert(h, 0);
printf("%d\n", pop(h));  // 0
```

Here is a `main` function you can use for testing:

```
int main() {
    MinHeap *h = createHeap();

    insert(h, 27);
    insert(h, 89);
    insert(h, 29);
    insert(h, 13);
    insert(h, 83);
    insert(h, 48);
    insert(h, 9);
    insert(h, 50);
    insert(h, 2);
    insert(h, 26);

    for (int i = 0; i < 11; i++) {
        printf("%d\n", pop(h));
    }

    free(h->data);
    free(h);
    return 0;
}
```

The expected output is:

```
2
9
13
26
27
29
48
50
83
89
-1
```