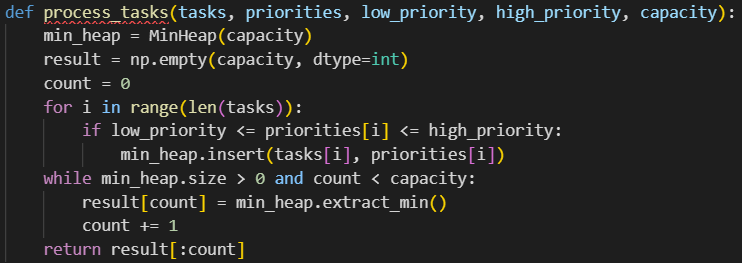
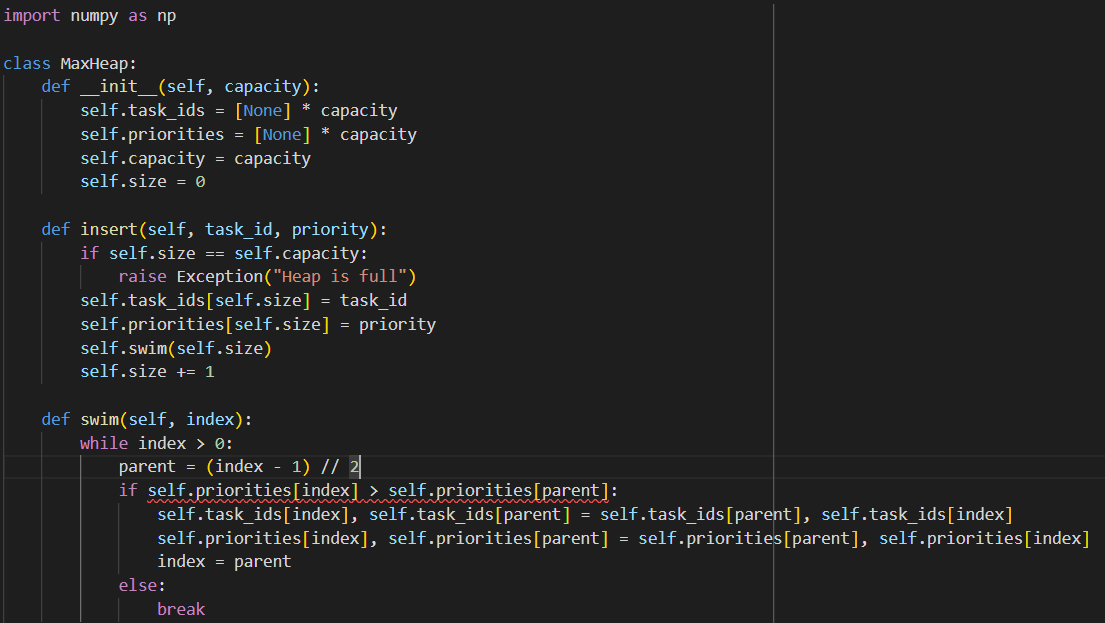
**SET A**

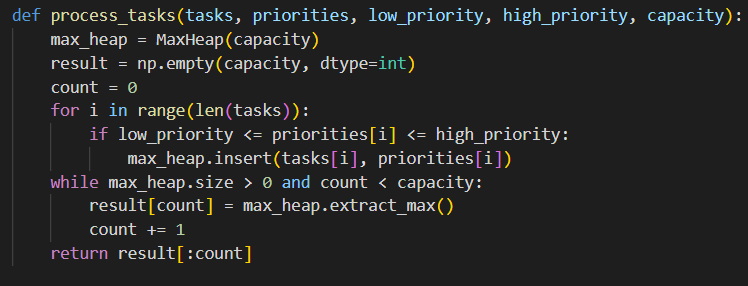


**Rubric**

* *2.5 Marks:* Construct the MinHeap class
* *2.5 Marks:* **insert()** -> Checks for proper addition of tasks, priority including the condition to handle a full heap.
* *1 Marks:* **swim()** -> Correctly calculated parent index
* *2 Marks:* **swim()** -> Check that the min-heap order is kept during insertions by swapping indexes correctly for both tasks and priority.
* *2 Marks:* **process\_tasks()** -> MinHeap is correctly initialized with the given capacity.
* *3 Marks:* **process\_tasks()** -> Check task within the range is inserted.
* 2 Mark: **process\_tasks()** -> Extract the minimum value from the heap and insert it in the result array.

**SET B**

****

****

**Rubric**

* *2.5 Marks:* Construct the MaxHeap class
* *2.5 Marks:* **insert()** -> Checks for proper addition of tasks, priority including the condition to handle a full heap.
* *1 Marks:* **swim()** -> Correctly calculated parent index
* *2 Marks:* **swim()** -> Check that the max-heap order is kept during insertions by swapping indexes correctly for both tasks and priority.
* *2 Marks:* **process\_tasks()** -> MaxHeap is correctly initialized with the given capacity.
* *3 Marks:* **process\_tasks()** -> Check task within the range is inserted.
* 2 Mark: **process\_tasks()** -> Extract the maximum value from the heap and insert it in the result array.