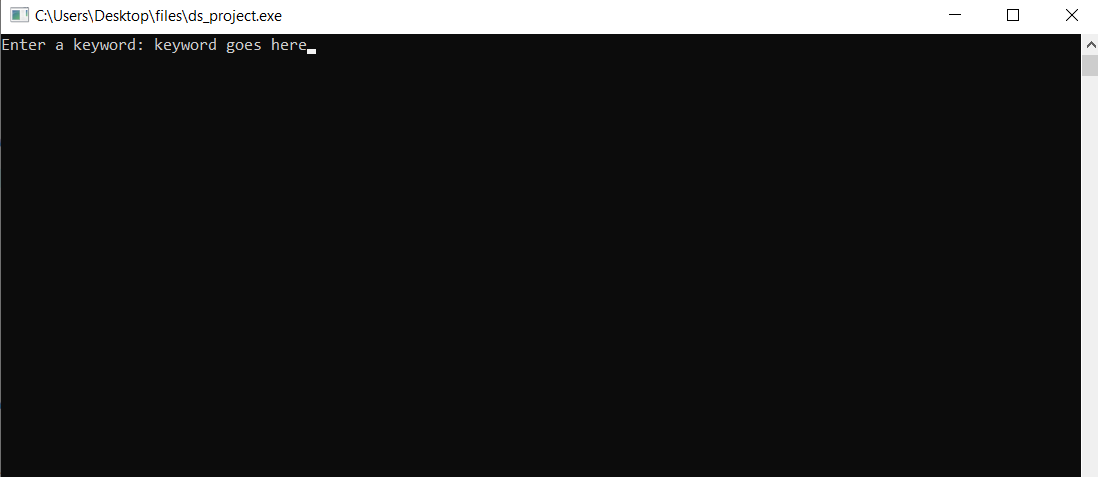
**CSE225 Data Structures Project II Report**

****

**Isim soyisimler**

**Question 1**

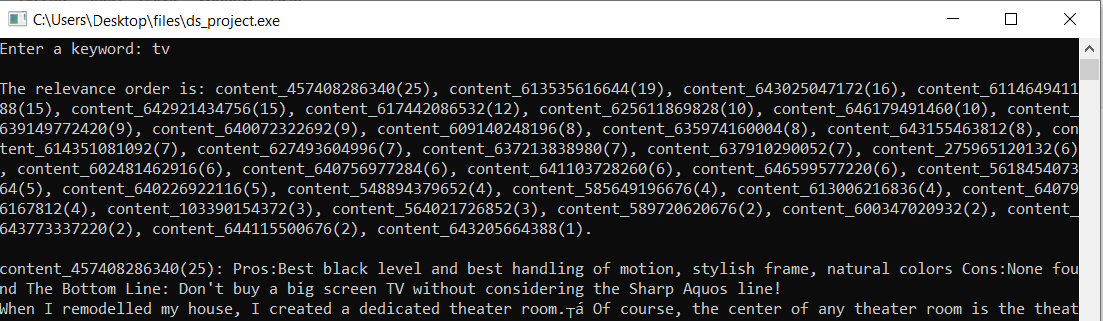
**Getting keyword input from user:**

****

**Relevant source code:**

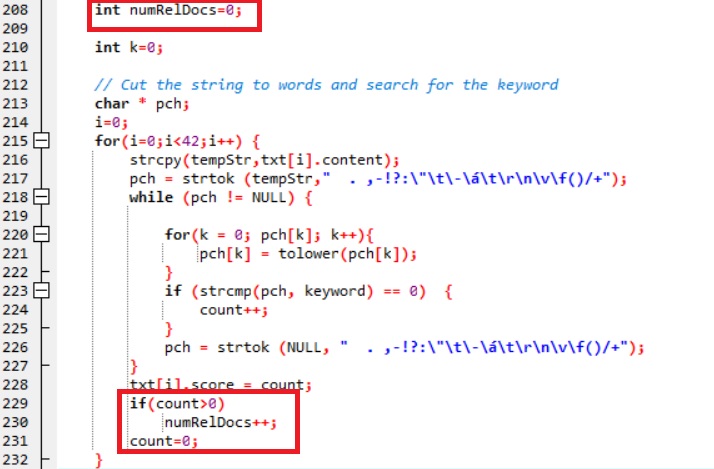
****

**Question 2**

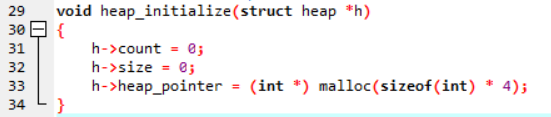
**Example run that shows number of relevant documents are calculated:**

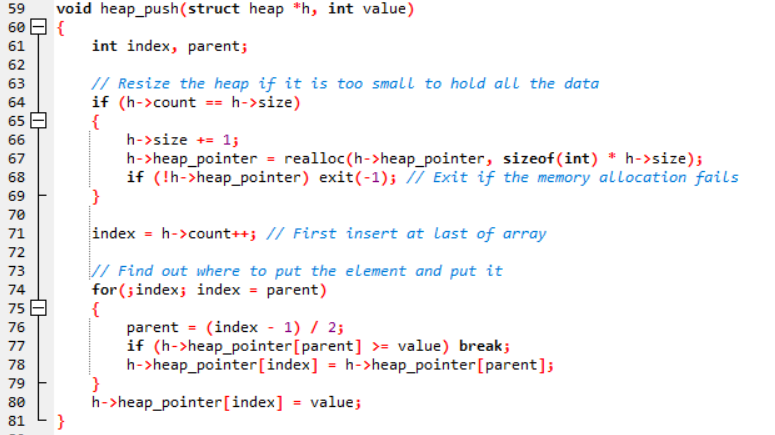
**In the image above, it can be seen that for the keyword “tv” relevant documents are taken into account until there is at least one match.**

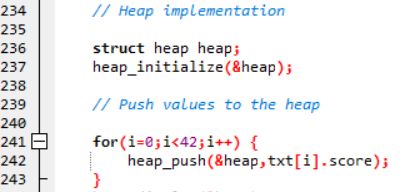
**Image below shows the piece of code where number of relevant documents are calculated in the variable numRelDocs. numRelDocs is incremented when there is at least one keyword match is found in a document.**

****

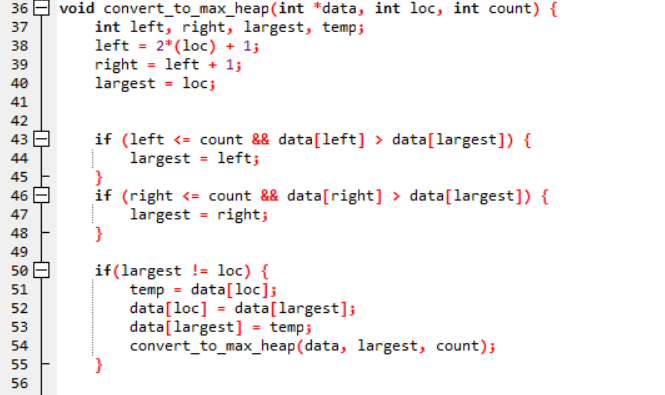
**Question 3**

**Enqueue implementation:**

**Function heap\_initialize initializes the heap.**

**Function heap\_push receives integer value to be pushed into the heap. Operations such as allocating needed space is handled here.**

**Above image shows how the functions are called from the main method. Notice how each relevance score is pushed into the heap.**

****

**Function convert\_to\_max\_heap rearranges heap structure as new nodes are added to max heap standards.**

**Dequeue implementation:**

****

**Print\_PrQ function is called from main method to start the dequeue process. Notice that the second argument takes number of relevant documents.**

****

**Function print\_PrQ handles priority queue extraction from the heap as well as console printing format. Two helper functions heap\_delete() [returns max node and deletes it from the heap] and searchDocID() [searches for the document ID that corresponds with match value and returns it] are used within for calculations.**

**Question 4**

**Here are a few test cases and their outputs. For convenience, contents to be printed after the relevance order is not included in the report, but the program prints it successfully.**

* **keyword = text**

Enter a keyword: text

The relevance order is: content\_585649196676(2).

* **keyword = tv**

Enter a keyword: tv

The relevance order is: content\_457408286340(25), content\_613535616644(19), content\_643025047172(16), content\_611464941188(15), content\_642921434756(15).

* **keyword = pros**

Enter a keyword: pros

The relevance order is: content\_640796167812(2), content\_103390154372(1), content\_241371942532(1), content\_275965120132(1), content\_457408286340(1).

* **keyword = size**

Enter a keyword: size

The relevance order is: content\_457408286340(5), content\_611464941188(3), content\_585649196676(2), content\_548894379652(1), content\_589720620676(1).

* **keyword = screen**

Enter a keyword: screen

The relevance order is: content\_457408286340(21), content\_613535616644(5), content\_275965120132(3), content\_617442086532(3), content\_639149772420(3).

* **keyword = buy**

Enter a keyword: buy

The relevance order is: content\_457408286340(7), content\_646599577220(4), content\_643025047172(3), content\_613006216836(2), content\_635974160004(2).

* **keyword = channel**

Enter a keyword: channel

The relevance order is: content\_548894379652(6), content\_457408286340(1), content\_602481462916(1), content\_609140248196(1), content\_611464941188(1).

* **keyword = samsung**

Enter a keyword: samsung

The relevance order is: content\_643773337220(10), content\_614351081092(9), content\_640226922116(6), content\_602481462916(4), content\_625611869828(4).