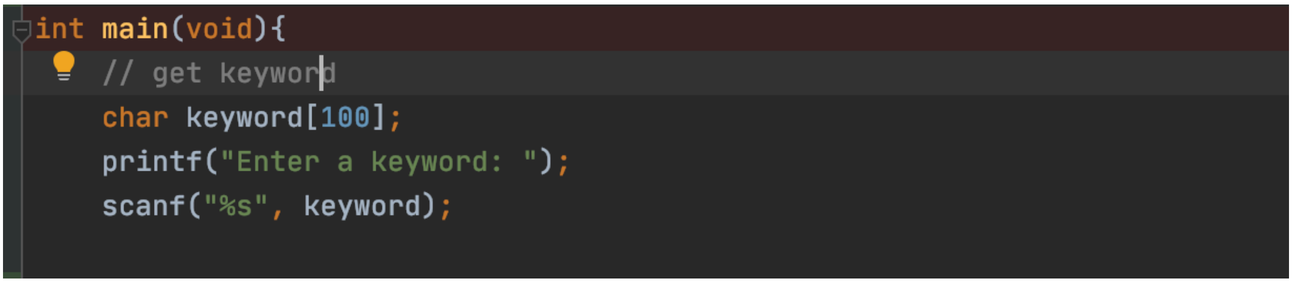
CSE225 Data Structures
Project 2 Report

Berkay Mengunogul
150119934

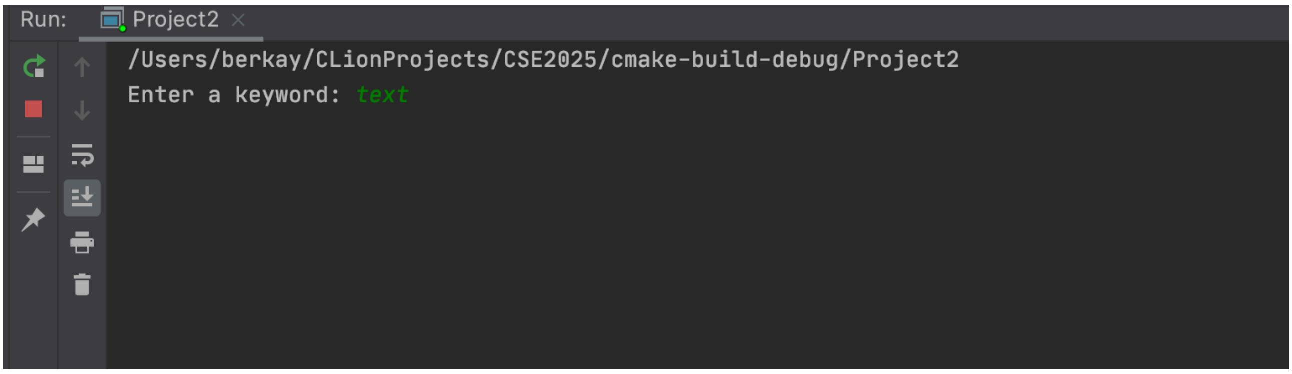
1) The keyword is taken from user is shown below.

From code:



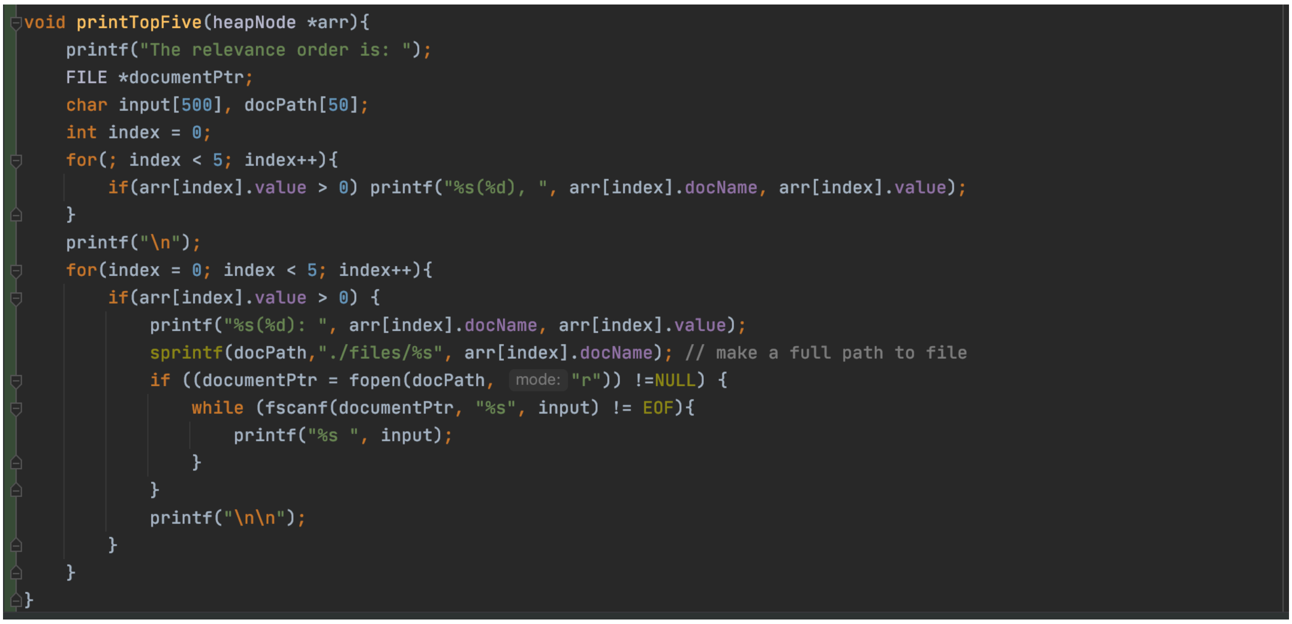
```
int main(void){
    // get keyword
    char keyword[100];
    printf("Enter a keyword: ");
    scanf("%s", keyword);
```

From console:



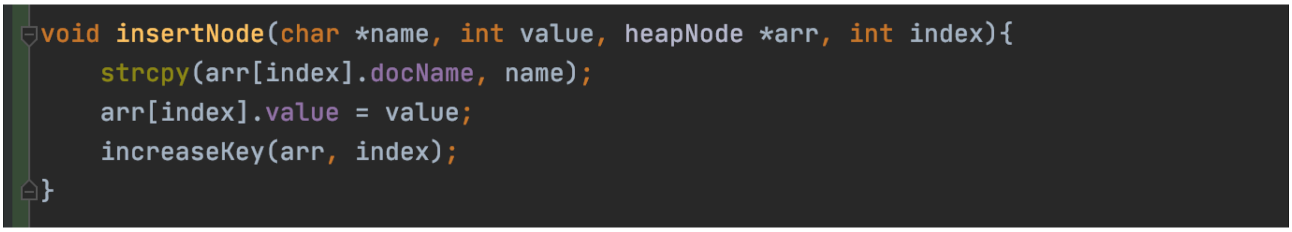
```
/Users/berkay/CLionProjects/CSE2025/cmake-build-debug/Project2
Enter a keyword: text
```

2) Since Max-Heap algorithm is used in this project. From the represented array first five nodes are taken as relevant documents. First for loop is for printing the found relevant document. Second for loop is for extracting the document and printing on console with documents’s name.



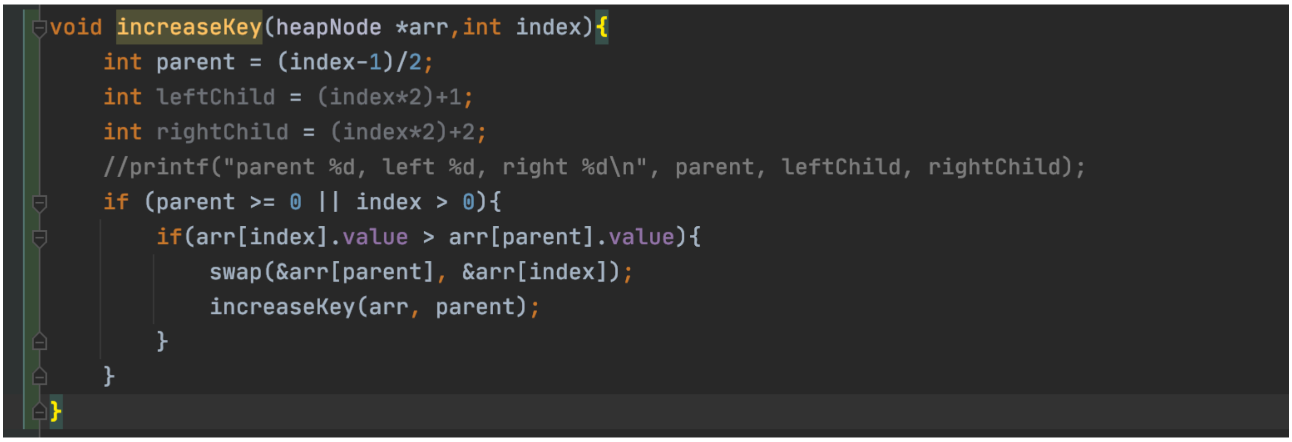
```
void printTopFive(heapNode *arr){
    printf("The relevance order is: ");
    FILE *documentPtr;
    char input[500], docPath[50];
    int index = 0;
    for(; index < 5; index++){
        if(arr[index].value > 0) printf("%s(%d), ", arr[index].docName, arr[index].value);
    }
    printf("\n");
    for(index = 0; index < 5; index++){
        if(arr[index].value > 0) {
            printf("%s(%d): ", arr[index].docName, arr[index].value);
            sprintf(docPath,"./files/%s", arr[index].docName); // make a full path to file
            if ((documentPtr = fopen(docPath,  mode: "r")) !=NULL) {
                while (fscanf(documentPtr, "%s", input) != EOF){
                    printf("%s ", input);
                }
            }
            printf("\n\n");
        }
    }
}
```

3) To my enqueue function(insertNode), name of the document, its value, represented array and it is index given. Index is the last element + 1 in here. With the given parameters, it is inserted at the end of array. From this part it is position is updated with increaseKey function.



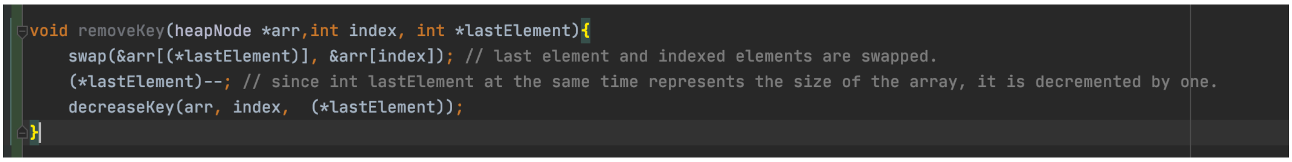
```
void insertNode(char *name, int value, heapNode *arr, int index){
    strcpy(arr[index].docName, name);
    arr[index].value = value;
    increaseKey(arr, index);
}
```

At increaseKey function, value of parent node and node itself compared. If value itself bigger than parent node then they are swapped and then function itself called again to check the position recursively.



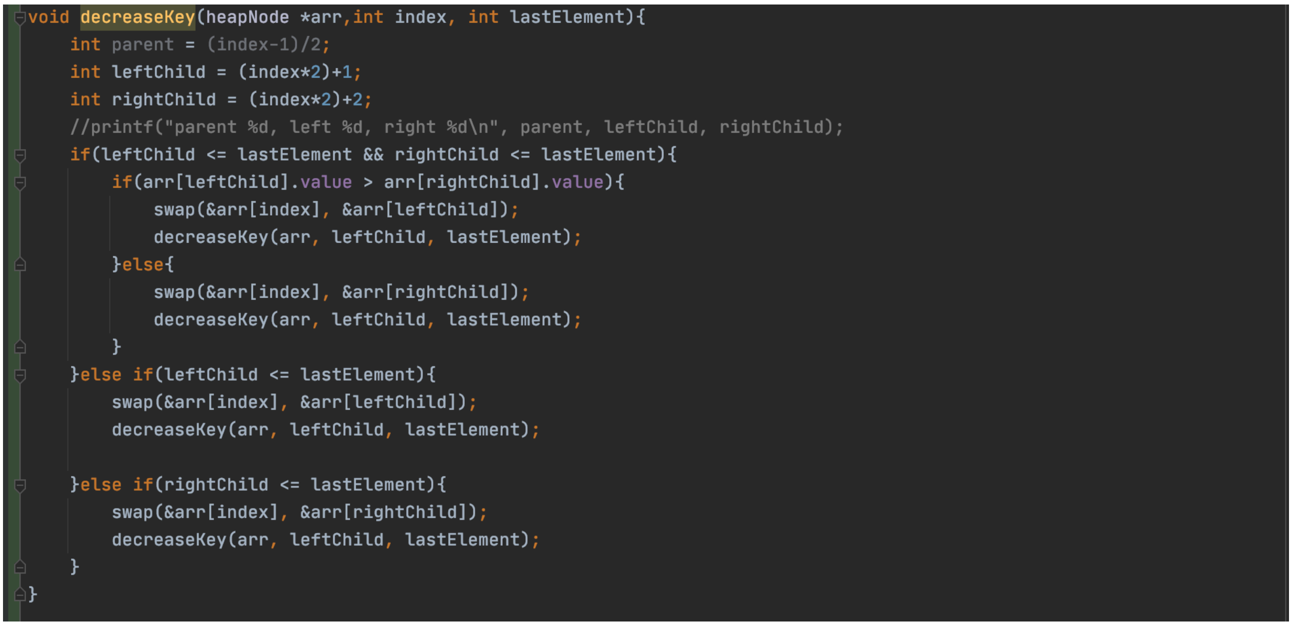
```
void increaseKey(heapNode *arr,int index){
    int parent = (index-1)/2;
    int leftChild = (index*2)+1;
    int rightChild = (index*2)+2;
    //printf("parent %d, left %d, right %d\n", parent, leftChild, rightChild);
    if (parent >= 0 || index > 0){
        if(arr[index].value > arr[parent].value){
            swap(&arr[parent], &arr[index]);
            increaseKey(arr, parent);
        }
    }
}
```

To my dequeue function(removeKey), represented array, index of the node and index of last element given. With the given parameters node is swapped with last element and since index of last element also represents the size of array it is decremented by one. For finding the position of swapped node, with it is index and index of last element decreaseKey Function called.



```
void removeKey(heapNode *arr,int index, int *lastElement){
    swap(&arr[(*lastElement)], &arr[index]); // last element and indexed elements are swapped.
    (*lastElement)--; // since int lastElement at the same time represents the size of the array, it is decremented by one.
    decreaseKey(arr, index,  (*lastElement));
}
```

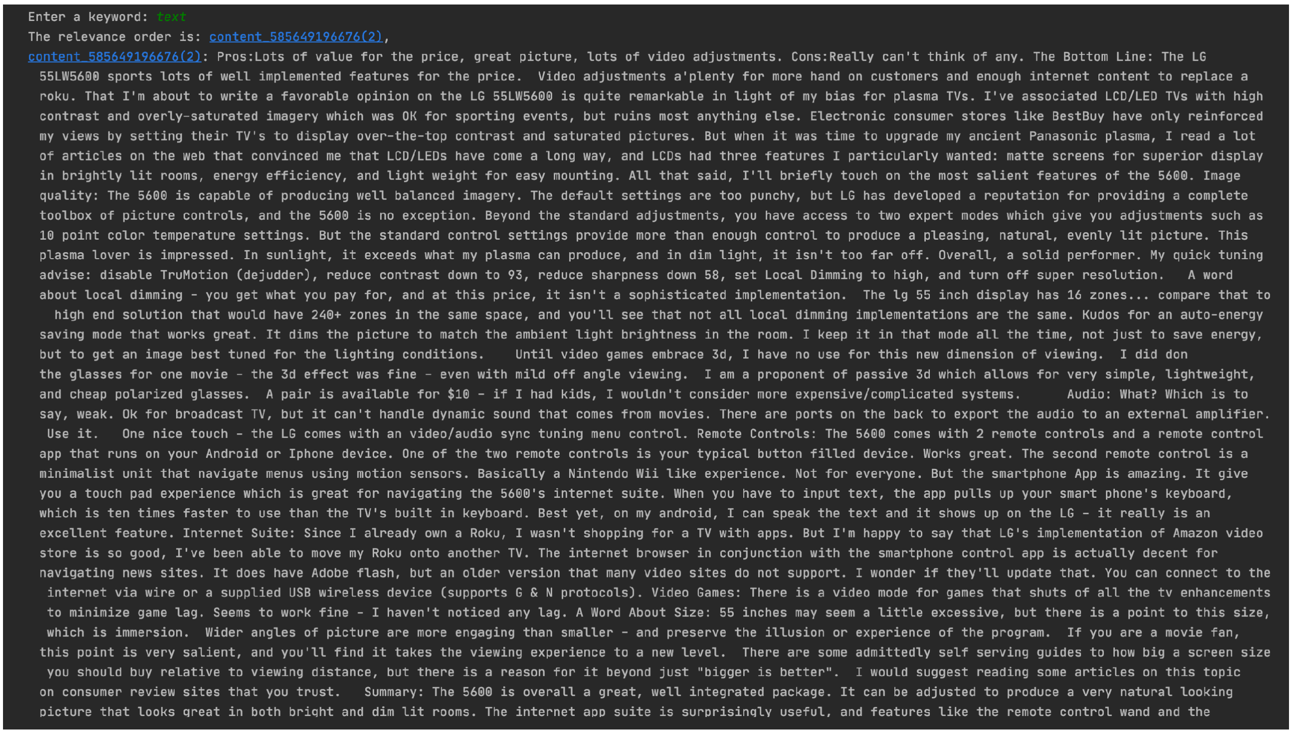
At decreaseKey function first, index of the child nodes checked whether they in range of array or not. If both child in range of array then value with the bigger node is swapped. If one of the them in range then the existing one swapped. Then function itself called again to check the position again recursively.



```
void decreaseKey(heapNode *arr,int index, int lastElement){
    int parent = (index-1)/2;
    int leftChild = (index*2)+1;
    int rightChild = (index*2)+2;
    //printf("parent %d, left %d, right %d\n", parent, leftChild, rightChild);
    if(leftChild <= lastElement && rightChild <= lastElement){
        if(arr[leftChild].value > arr[rightChild].value){
            swap(&arr[index], &arr[leftChild]);
            decreaseKey(arr, leftChild, lastElement);
        }else{
            swap(&arr[index], &arr[rightChild]);
            decreaseKey(arr, leftChild, lastElement);
        }
    }else if(leftChild <= lastElement){
        swap(&arr[index], &arr[leftChild]);
        decreaseKey(arr, leftChild, lastElement);

    }else if(rightChild <= lastElement){
        swap(&arr[index], &arr[rightChild]);
        decreaseKey(arr, leftChild, lastElement);
    }
}
```

4) There is only one matching document for “text” word it 2 times occurrence. The output is given below.



```
Enter a keyword: text
The relevance order is: content_585649196676(2),
content_585649196676(2): Pros:Lots of value for the price, great picture, lots of video adjustments. Cons:Really can't think of any. The Bottom Line: The LG
  55LW5600 sports lots of well implemented features for the price.  Video adjustments a'plenty for more hand on customers and enough internet content to replace a
  roku. That I'm about to write a favorable opinion on the LG 55LW5600 is quite remarkable in light of my bias for plasma TVs. I've associated LCD/LED TVs with high
  contrast and overly-saturated imagery which was OK for sporting events, but ruins most anything else. Electronic consumer stores like BestBuy have only reinforced
  my views by setting their TV's to display over-the-top contrast and saturated pictures. But when it was time to upgrade my ancient Panasonic plasma, I read a lot
  of articles on the web that convinced me that LCD/LEDs have come a long way, and LCDs had three features I particularly wanted: matte screens for superior display
  in brightly lit rooms, energy efficiency, and light weight for easy mounting. All that said, I'll briefly touch on the most salient features of the 5600. Image
  quality: The 5600 is capable of producing well balanced imagery. The default settings are too punchy, but LG has developed a reputation for providing a complete
  toolbox of picture controls, and the 5600 is no exception. Beyond the standard adjustments, you have access to two expert modes which give you adjustments such as
  10 point color temperature settings. But the standard control settings provide more than enough control to produce a pleasing, natural, evenly lit picture. This
  plasma lover is impressed. In sunlight, it exceeds what my plasma can produce, and in dim light, it isn't too far off. Overall, a solid performer. My quick tuning
  advise: disable TruMotion (dejudder), reduce contrast down to 93, reduce sharpness down 58, set Local Dimming to high, and turn off super resolution.   A word
  about local dimming - you get what you pay for, and at this price, it isn't a sophisticated implementation.  The lg 55 inch display has 16 zones... compare that to
    high end solution that would have 240+ zones in the same space, and you'll see that not all local dimming implementations are the same. Kudos for an auto-energy
  saving mode that works great. It dims the picture to match the ambient light brightness in the room. I keep it in that mode all the time, not just to save energy,
  but to get an image best tuned for the lighting conditions.    Until video games embrace 3d, I have no use for this new dimension of viewing.  I did don
  the glasses for one movie - the 3d effect was fine - even with mild off angle viewing.  I am a proponent of passive 3d which allows for very simple, lightweight,
  and cheap polarized glasses.  A pair is available for $10 - if I had kids, I wouldn't consider more expensive/complicated systems.      Audio: What? Which is to
  say, weak. Ok for broadcast TV, but it can't handle dynamic sound that comes from movies. There are ports on the back to export the audio to an external amplifier.
   Use it.   One nice touch - the LG comes with an video/audio sync tuning menu control. Remote Controls: The 5600 comes with 2 remote controls and a remote control
  app that runs on your Android or Iphone device. One of the two remote controls is your typical button filled device. Works great. The second remote control is a
  minimalist unit that navigate menus using motion sensors. Basically a Nintendo Wii like experience. Not for everyone. But the smartphone App is amazing. It give
  you a touch pad experience which is great for navigating the 5600's internet suite. When you have to input text, the app pulls up your smart phone's keyboard,
  which is ten times faster to use than the TV's built in keyboard. Best yet, on my android, I can speak the text and it shows up on the LG - it really is an
  excellent feature. Internet Suite: Since I already own a Roku, I wasn't shopping for a TV with apps. But I'm happy to say that LG's implementation of Amazon video
  store is so good, I've been able to move my Roku onto another TV. The internet browser in conjunction with the smartphone control app is actually decent for
  navigating news sites. It does have Adobe flash, but an older version that many video sites do not support. I wonder if they'll update that. You can connect to the
   internet via wire or a supplied USB wireless device (supports G & N protocols). Video Games: There is a video mode for games that shuts of all the tv enhancements
   to minimize game lag. Seems to work fine - I haven't noticed any lag. A Word About Size: 55 inches may seem a little excessive, but there is a point to this size,
   which is immersion.  Wider angles of picture are more engaging than smaller - and preserve the illusion or experience of the program.  If you are a movie fan,
  this point is very salient, and you'll find it takes the viewing experience to a new level.  There are some admittedly self serving guides to how big a screen size
   you should buy relative to viewing distance, but there is a reason for it beyond just "bigger is better".  I would suggest reading some articles on this topic
  on consumer review sites that you trust.   Summary: The 5600 is overall a great, well integrated package. It can be adjusted to produce a very natural looking
  picture that looks great in both bright and dim lit rooms. The internet app suite is surprisingly useful, and features like the remote control wand and the
```

5) Priority Queues are such structures that can shorten the average waiting time of process. Let’s assume in a queue, two different datas enqueued at the same time.If first one will take 10 seconds and the second will take 2 seconds. Then in FIFO queues first process will respond in 10 secs but second process will respond in 12 seconds. This will cause 11 secs average waiting time. In the other hand in min-heap structured priority queues first shorter process will respond in 2 secs then other process will respond 12 seconds. So, average waiting time will be 6 seconds. It is a huge advantage.