Computer Organization Project#1 Report

Question 1. Build a min-heap
When you run the program, a welcoming menu shows off and waits for user's input. There are four options. Insert, Delete, Print and Exit.

```
Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:
```

1.a - Insert

In the insertion part, firstly, the algorithm takes A[0] and check if it is equal to maximum heap size. If not, prints a message to get an input and reads it. After reading it, places this input value to the index A[0] +1 which is the next empty place in the array. After inserting the value, program checks if it is smaller than it's parent in a while loop until it gets to right place. If it is smaller than it's parent, basically they swap. If not, loop finishes.

```
Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:1

Please enter the value of item:12
The value of 12 is placed at location 1 of the Heap Array!
Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:1

Please enter the value of item:3
The value of 3 is placed at location 1 of the Heap Array!
Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:1

Please enter the value of item:5
The value of 5 is placed at location 3 of the Heap Array!
Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:
```

1.b - Delete Min

This function deletes the minimum element (root) of min heap. The algorithm works in this way: First, we swap smallest element with last one, then reduce the number of elements in the heap. After this procedure is done, we have to keep min-heap property. To do that, starting from root, we check every parent with it's childs. Firstly, we check whether if parent's left child's index is bigger than heap size, then we compare their values. If parent is bigger than child, we mark left child as smaller. After this, we compare right child's index number is greater than heap size. If it is not, we compare smallest's value with right child's value. If right child is smaller, we mark it as smallest, otherwise smallest doesn't change. After these are done, we lastly compare smallest with parent's index. If they are equal, that means parent is smaller than it's childs and we do not need to swap them. If they are not, we swap parent with smallest and set current smallest value as new parent to continue to keeping min-heap property.

```
The size of heap is 7
[1] 10
[2] 20 - [3] 30
[4] 40 - [5] 50     [6] 60 - [7] 70

Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:2
Deleting smallest value.

Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:3
The size of heap is 6
[1] 20
[2] 40 - [3] 30
[4] 70 - [5] 50     [6] 60 - X
```

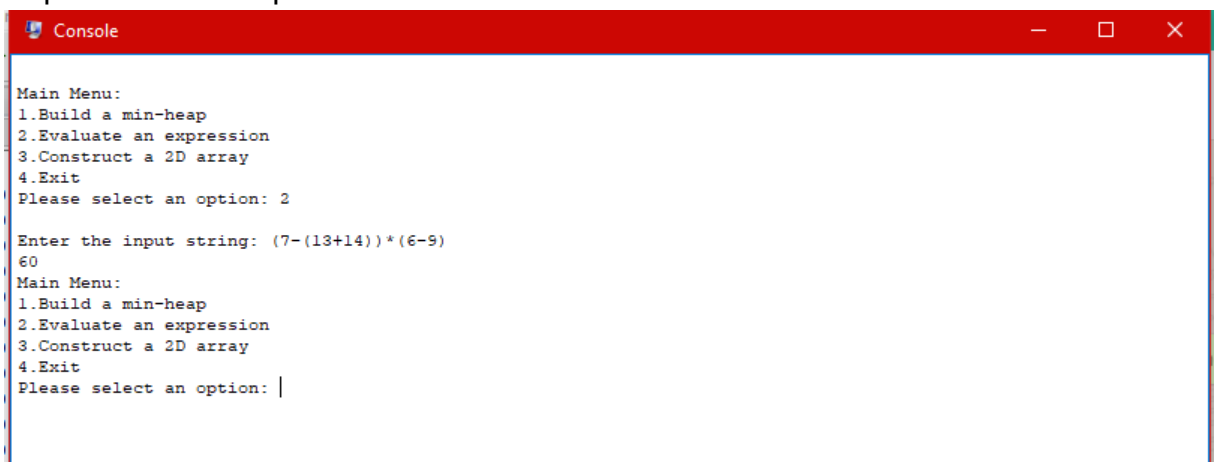
1.c - Print Heap

Firstly, we check if the heap is empty or not. If it is, finish printing. Else, check if all elements in the array are printed or not. If they are, print X. Else , print the index of the element between brackets. Then, print the value in that index. If the next index is sibling of current one, we print a dash between them. Else we print tab. If the printed number of values in a row is equal to max number of values in that line, we print \n then continue.

```
Choose operation:
1.Insert an item to the heap.
2.Delete an item from the heap.
3.Print the heap.
4.Return back
Please select an option:3
The size of heap is 4
[1] 1
[2] 5 - [3] 10
[4] 12 - X     X - X
```

2- In this problem we used stack. First of all, we begin to scan the string character by character. In the string there are numbers, math operations, open parenthesis, close parenthesis. We pushed numbers and math operations but, when we see close parenthesis pop 2 numbers and operation do the calculation and push the result into the stack again. When string is finished check the stack twice first look at multiplications and divisions, do the calculations push result and rearrange the stack. Second time look at additions and subtractions at the end, the result will be pushed.

Pop the result and print to secreen.



3- First, we check the string character by character, conver the character into integer when we see space character we insert the number into the 2D array. For displaying 2D array we calculated number of column, number of elements divided with row number if there is a remainder increase column number by 1 and get correct result. Need two variables 1 is for column 1 is for row when element is written increase column if it is equal NumberOfColumn clear column variable and increase row variable if row variable is equal NumberOfColun display is done. For the convert decimal to hexadecimal need string divide address by 16 and convert remainder into hexadecimal form then insert it into 8th byte of string then divide by 16 again and insert remainder 7th byte of string do it untill number is less than 16. For max and min value look all number between A[0][0] and A[row][column] using this formula

Address = baseaddress + [rowIndex * columnSize + ColumnIndex] * dataSize