Part 1: Pick the best answer to each of the following questions. Enter your choice into the space provided to the left of the question number. (Total 10 points)

______1. A "linked list" is:

A An arithmetic series.

B A compiler directive.

C An implementation of a list that uses a linked data structure.

D None of the above.

______2. The postfix expression `14 2 5 + =` will generate an error, because ____.

A it contains an illegal operator

B it does not have enough operands

C it has too many operators

D there will be too many elements in the stack when the equal sign is encountered

______3. A "heap" is a:

A A first in first out collection

B A type of asymptotic bound

C A binary tree in which *parent data* <= *children data*

D None of the above

______4. For Mergesort, the Big O notation is:

A O(n*log$n$)

B O(log$n$)

C O(n^2)

D O(n)

______5. What is the output of the following code:

```
a Stack<int> stack;
int x, y;
x = 5;
y = 3;
stack.push(8);
stack.push(x);
stack.push(y + 1);
y = stack.top();
stack.pop();
stack.push(x + y);
x = stack.top();
stack.pop();
System.out.print("x = " + x + " ");
System.out.println("y = " + y);
```

A `x = 5  y = 3`

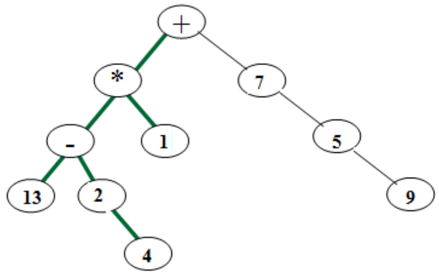
B `x = 9  y = 4`

C `x = 11 y = 4`

D `Other; specify x =   y =`

Part 2: Write answer to the following questions. (Total 60 points)

Question 1. Please show the different traversal output on the following tree: **(10 pts)**



(a) Inorder traversal:

13 – 2 4 * 1 + 7 5 9

(b) Postorder traversal:

13 4 2 – 1 * 9 5 7 +

(c) Preorder traversal:

+ * - 13 2 4 1 7 5 9

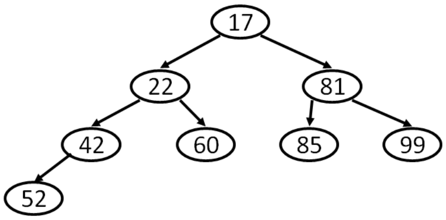
(d) Breadth first traversal:

+ * 7 - 1 5 13 2 9 4

(e) Depth first traversal:

+ * - 13 2 4 1 7 5 9

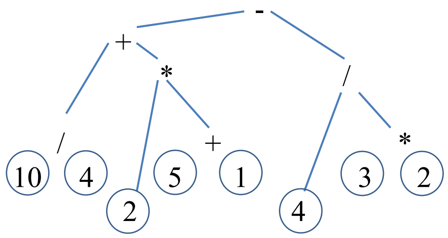
Question 2. Show the steps of Heapsort on the following minHeap to sort the data in **descending** order. Use the tables provided below to show the status of the array after each iteration of Heapsort. **(10 pts)**



1	17	22	81	42	60	85	99	52
2	52	22	81	42	60	85	99	**17**
3	22	42	81	52	60	85	99	17
4	99	42	81	52	60	85	**22**	**17**
5	42	52	81	99	60	85	22	17
6	85	52	81	99	60	**42**	**22**	**17**
7	Continue							
8	Continue							

Question 3. Consider the following infix expression: $10/4\ +\ 2 * (5 + 1)\ -\ (4/(3 * 2))$ **(8 pts)**

a) Draw an expression tree



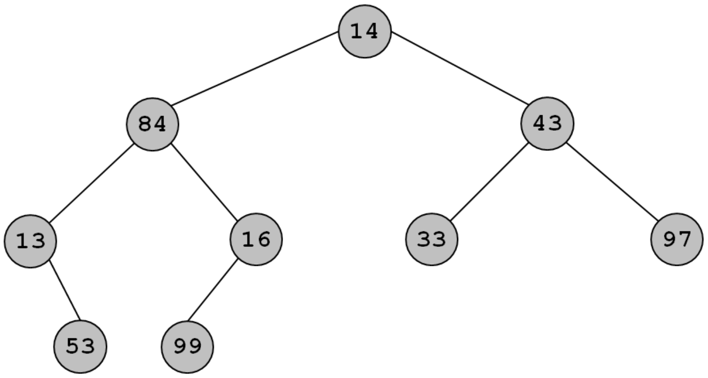
b) Write the polish notation (perform a preorder traversal of the expression tree) and evaluate

- + / 10 4 * 2 + 5 1 / 4 * 3 2

c) Write the reverse polish notation (perform a postorder traversal of the expression tree) and evaluate

10 4 / 2 5 1 + * + 4 3 2 * / -

Question 8. Recall the non-recursive preorder traversal algorithm for a binary tree. Do a preorder traversal of the binary tree given below. Show the stack contents after each push and pop operation. You can use the output bar and the stack containers provided below to show your output. If needed, you can draw more containers. **(8 pts)**



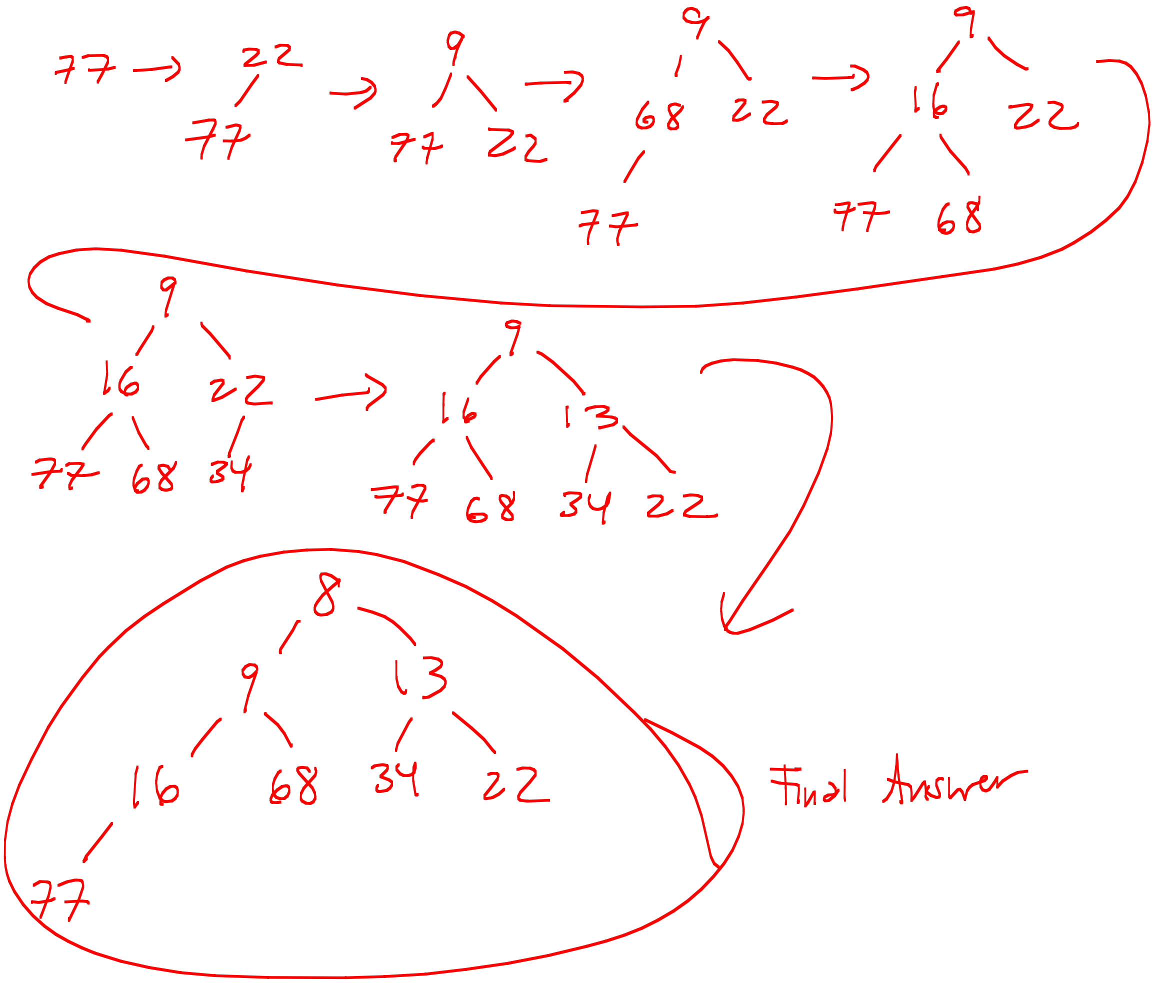
Output	Output 14	Output 14 84	Output 14 84 13
14	84 43	13 16 43	53 16 43
Stack	Stack	Stack	Stack

Output 14 84 13 53	Output 14 84 13 53 16	Output 14 84 13 53 16 99 43	Output 14 84 13 53 16 99 43 33
16 43	99 43	33 97	97
Stack	Stack	Stack	Stack

Output 14 84 13 53 16 99 43 33 97	Output	Output	Output
Stack	Stack	Stack	Stack

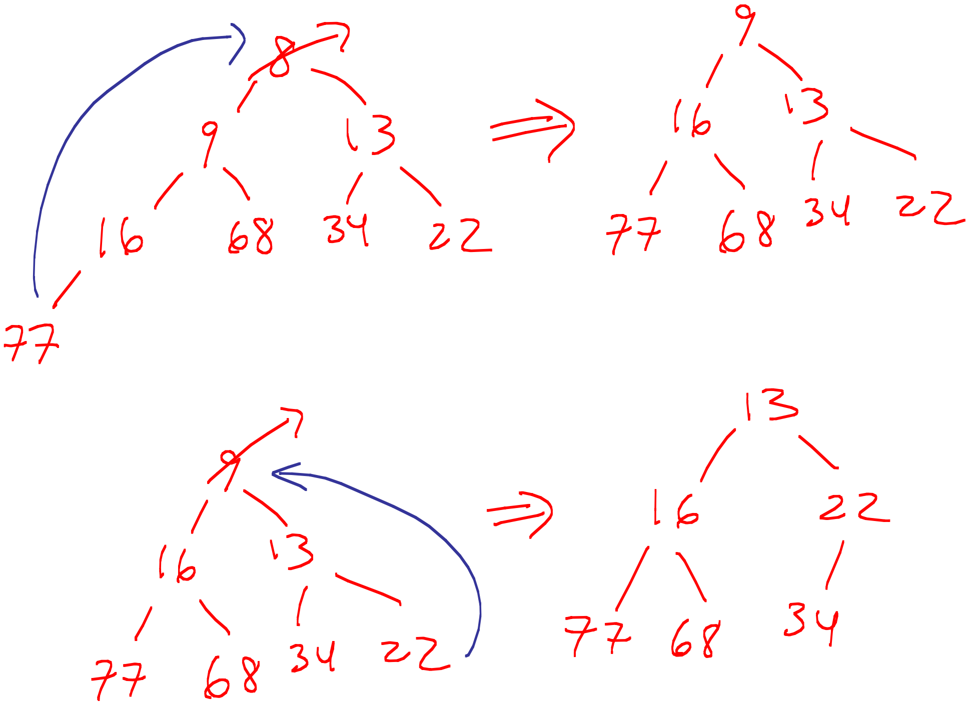
4) Heaps

a) Draw the binary min heap that results from inserting: 77, 22, 9, 68, 16, 34, 13, 8 in that order into an initially empty binary min heap. You do not need to show the array representation of the heap. You are only required to show the final heap, although if you draw intermediate heaps, ***please circle your final result for ANY credit***.



Final Answer

b) Draw the binary min heap that results from doing 2 deletemins on the heap you created in part a). You are only required to show the final heap, although if you draw intermediate heaps **please circle your final result for ANY credit**.



c) What is the null path length of the root node in the last heap you drew in part b) above?

one

2. (8 pts) **Big-Oh and Run Time Analysis:** Describe the worst case running time of the following pseudocode functions in Big-Oh notation in terms of the variable `n`. ***Showing your work is not required*** (although showing work ***may*** allow some partial credit in the case your answer is wrong – don't spend a lot of time showing your work.). You MUST choose your answer from the following (not given in any particular order), each of which could be re-used (could be the answer for more than one of I. – IV.):

   $O(n^2)$, $O(n^3 \log n)$, $O(n \log n)$, $O(n)$, $O(n^2 \log n)$, $O(n^5)$, $O(2^n)$, $O(n^3)$, $O(\log n)$, $O(1)$, $O(n^4)$, $O(n^n)$

```
 I. void silly(int n) {
        for (int i = 0; i < n; ++i) {
           j = n;
           while (j > 0) {
              System.out.println(″j = ″ + j);
              j = j - 2;
           }
        }
    }
```

Runtime:

$O(n^2)$

```
II. void silly(int n, int x, int y) {
        for (int k = n; k > 0; k--)
           if (x < y + n) {
              for (int i = 0; i < n; ++i)
                 for (int j = 0; j < i; ++j)
                    System.out.println(″y = ″ + y);
           } else {
              System.out.println(″x = ″ + x);
           }
    }
```

$O(n^3)$

```
III.   void silly(int n) {
          for (int i = 0; i < n; ++i) {
                for (int j = 0; j < n; ++j)
                   System.out.println(″j = ″ + j);
                for (int k = 0; k < i; ++k) {
                   System.out.println(″k = ″ + k);
                   for (int m = 0; m < 100; ++m)
                      System.out.println(″m = ″ + m);
                 }
          }
       }
```

$O(n^2)$

```
 IV.
    int silly(int n, int m) {
        if (m < 2) return m;
        if (n < 1) return n;
        else if (n < 10)
          return silly(n/m, m);
        else
          return silly(n - 1, m);
      }
```

$O(n)$