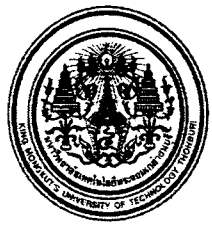
Key 31 พ.ค. 61.

Seat No.



King Mongkut's University of Technology Thonburi
Final Exam of Second Semester, Academic Year 2017

COURSE
CPE 113 Algorithms and Data Structures — **Computer Engineering 4th Yr.**
CPE 131 Programming with Data Structures — **Automation Engineering 1st Yr.**
Wednesday, May 16, 2018 — **13.00-16.00 h.**

Instructions

1. This examination contains 11 questions, 9 pages (including this cover page).
2. The answers must be written in the examination paper.
3. No books, notes, calculators or any other documents can be taken into the examination room.
4. Use your consideration and explain it if you have certain doubts about the exam questions.

Name-Lastname_____ Student ID #_____

Students must raise their hand to inform to the proctor upon their completion of the examination, to ask for permission to leave the examination room.
Students must not take the examination and the answers out of the examination room.

Students will be punished if they violate any examination rules. The highest punishment is dismissal.

Exam created by

Nuttanart Facundes
.....
(Asst. Prof. Dr. Nuttanart Facundes)

This examination has been approved by the committee of Computer Engineering Department

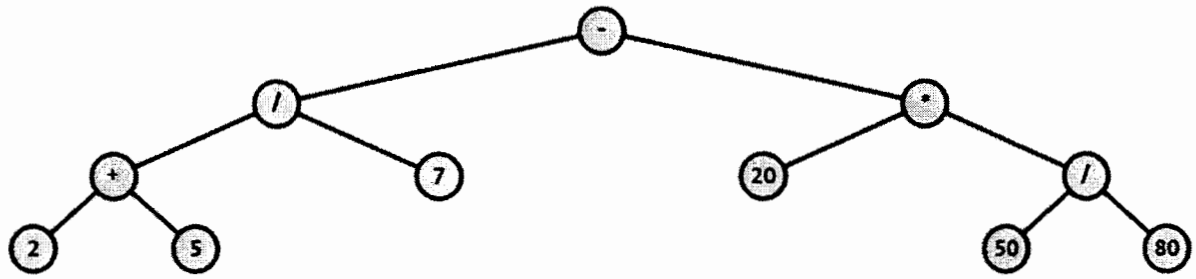
.....
(.....)
International Undergraduate Program Chairperson
Date.....10 MAY 2018.....

Name_____ ID#_____

Total points: 40 points (25% of grading)

1) (4 points)

The expression tree below is implemented by the binary tree data structure. Each of the tree node stores the array of character (string) and its children. Answer the following questions:




```
typedef struct _nodetree {
      char data[10];
      struct _nodetree * leftChild;
      struct _nodetree * rightChild;
} NODETREE;
NODETREE * root;
```

A) What kind of the tree traversal is matched with the following C codes (Pre-, Post- or In-order)?

```
void traversal_1 (Node * current) {
     if(hasLeftChild(node)) { traversal_1(node->leftChild); }
     printf("%s", node->data);
     if(hasRightChild(node)) { traversal_1(node->rightChild); }
}
```

Answer _____

B) What kind of the tree traversal is matched with the following C codes (Pre-, Post- or In-order)?

```
void traversal_2 (Node * current) {
     printf("%s ", node->data);
     if(hasLeftChild(node)) traversal_2 (node->leftChild);
     if(hasRightChild(node)) traversal_2 (node->rightChild);
}
```

Answer _____

C) If the root variable stores the pointer of the root node of the given tree (minus sign), what is the result of the *traversal_2(root);*?

Answer _____

Name_____ ID#_____

2) (5 points)

Given the EMPLOYEE data structure in C language and the pseudocode of bubble sort below.

```
typedef struct _employee {
    char name[50];
    char employee_type[30]; // {full-time, part-time}
    double salary;
} EMPLOYEE;
```

```
Algorithm bubblesort (array, count)
   set current to 0
   set sorted to FALSE
   while current < count and sorted = FALSE
        set walker to (count - 1)
        set sorted to TRUE
        while walker > current
            if array[walker] < array[walker-1]
                set sorted to FALSE
                swap the data in array at walker and(walker-1)
            decrease walker by 1
        increase current by 1
```

Suppose that *there is already the function to swap two employees*, so you can call it e.g. swap(array, index1, index2).

Write down the C code for bubble sorting the array of employees by salary in ascending order.

```
void bubblesort (EMPLOYEE array[100], int count) {









}
```

Name_____ ID#_____

3) (3 points)
There are many searching algorithms to find an item in the array of data. The result of the searching algorithm is the index whose key (or data to search) is found first in the array. However, the result could be -1 if there is no data that is equals to the key.

This question is about the binary search which is implemented by loop. The algorithm operates by comparing the key with the middle data of the sorted data array. The comparison result could be one of three cases: equal, lesser or higher. For each case, it is required to do something before the next round of the loop or to stop the loop.

Given that the array is sorted in ***descending order*** (from high to low). Write the C codes for each one of the three cases to complete the algorithm.

```
void binarySearch (int data[100], int low, int high, int key)
{
     while (low <= high) {
          int mid = (low + high) / 2;
          if (data[mid] == key) { _____<case1>_____ }
          if (data[mid] < key) { _____<case2>_____}
          if (data[mid] > key) { _____<case3>_____}
     }
     return -1;
}
```

Answer
case1 is _____
case2 is _____
case3 is _____

4) (2 points)
Given C code of some function declarations and the incomplete heapsort function as follow.

```
void heapify(int array[100], int childLoc);
void reheapUp(int array[100], int childLoc);
void reheapDown(int array[100], int size, int rootLoc);
void heapsort (int array[100], int size) {
     _____(1)_____
     int i;
     for(i=size-1; i>0; i--){
          swap(array, 0, i);
          _____(2)_____
     }
}
```

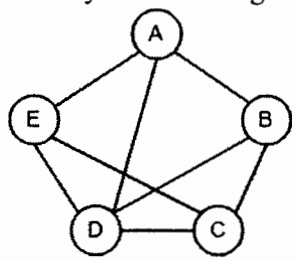
Name_____ ID#_____

Suppose that the functions *heapify*, *reheapUp* and *reheapDown* are already implemented. To correctly write the heapsort algorithm in C code, what is the best choice for (1) and (2)?

A	(1) heapify(array, 0); (2) reheapUp(array, i);	B	(1) heapify(array, 0); (2) reheapDown(array, i, 0);
C	(1) reheapUp(array, 0); (2) reheapDown(array, i, 0);	D	(1) reheapUp(array, 0); (2) heapify(array, 0);
E	(1) reheapDown(array, 0, 0); (2) heapify(array, i);	F	(1) reheapDown(array, 0, 0); (2) reheapUp(array, i);

Answer _____

5) (4 points)
This question is related to our lab about graph data structure. Given the graph that can visualized by the following figure and is represented by the following adjacency matrix



	A	*B*	*C*	*D*	*E*
A	0	1	0	1	1
B	1	0	1	1	0
C	0	1	0	1	1
D	1	1	1	0	1
E	1	0	1	1	0

A) Suppose we have the same implementation to our code in the lab session, if we remove Node B from graph, what will be the new adjacency matrix?

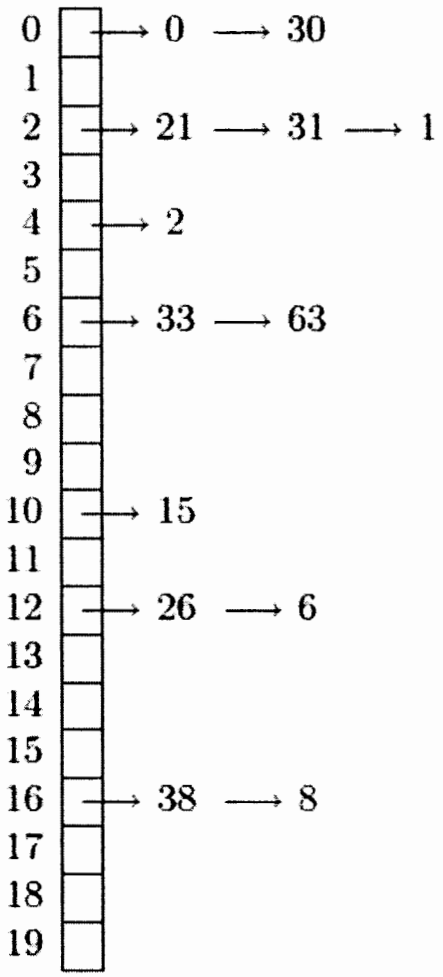
B) Continue from the previous question. If we add the new node that connects to all nodes excepts node D. what will be the newer adjacency matrix?

Name_____ ID#_____

6) Hashing (4 points)

Suppose you have a hash table and have inserted some elements. The results look as in the picture below. Answer the following questions:

a. What is the problem here?
b. What is the hash function used here?
c. What are the possible ways to solve the problem in 6a?



Name_____ ID#_____

7) (3 points)
Here is an array of 10 integers

5	3	8	9	1	7	0	2	6	4

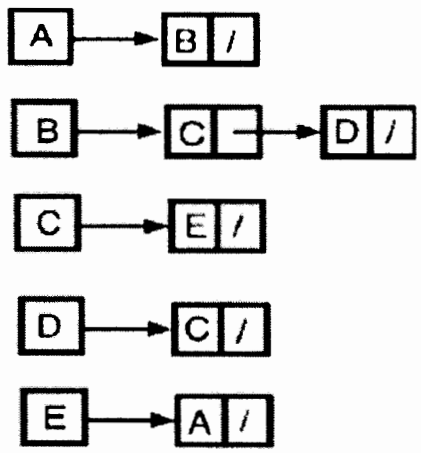
Suppose we partition this array using quicksort's partition function and use 5 for the pivot, what would the array be after the partition finishes?

8) Select True-False or select from multiple choices (3 points)

8.1 **T F** A heap can be a useful tool for sorting.
8.2 **T F** For Greedy Algorithms, we need to plan things many steps in advance.
8.3 An array of 7 integers is being sorted by the heapsort algorithm. After the initial phase of the algorithm (i.e. constructing the heap), which or the following is a possible ordering for the array?

A. 85 78 45 51 53 47 49
B. 85 49 78 45 47 51 53
C. 85 78 49 45 47 51 53
D. 45 85 78 53 51 49 47
E. 85 51 78 53 49 47 45

9) (5 points)
The adjacency list representation of a graph with 5 vertices: A, B, C, D, E is given below. Draw the corresponding adjacency matrix and graph.

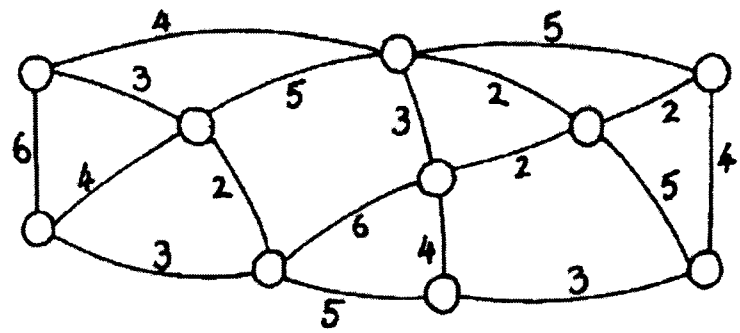


/ = null symbol

Name_____ ID#_____

Answer for 9)

10) Find the minimum spanning tree of the following graph. (5 points)



Name_____ ID#_____

10) Huffman Coding (2 points)
Construct the Huffman Tree using the symbols and the frequencies below:

The symbol frequencies are:

symbol	`e`	`l`	`n`	`s`
frequency	4/13	1/13	2/13	6/13

11) Bonus points (up to *2 points)*

Name **an algorithm** and **a data structure** that are your favorite (maybe they are useful, you used them in your project, etc.) Briefly describe them.