BBM201 – Data Structures – Fall 2017 - Final Exam
10.01.2018 – 120 minutes

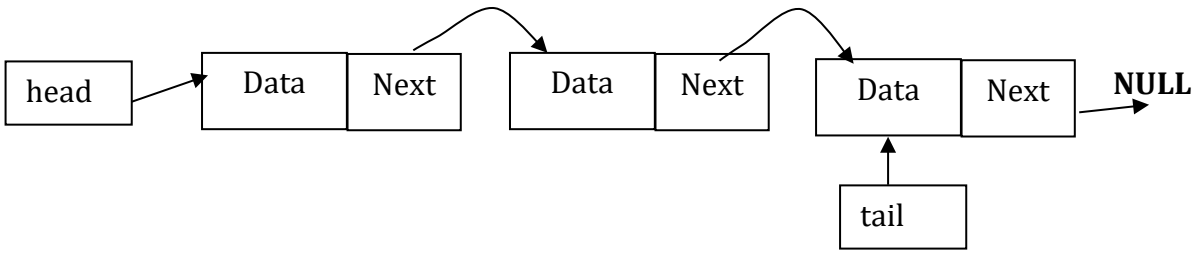
Name Surname: _____

Student ID : _____

Section: ☐ **Section 1 (Burcu Can)**
☐ **Section 2 (Sevil Şen)**
☐ **Section 3 (Adnan Özsoy)**

Questions	1	2	3	4	5	6	7	Total
Points	16	15	10	12	12	20	15	100
Grade								

1. (Linked List) ***(16 points)*** 1. Consider the linked list shown below. Please write the function named ***removeTail*** that will remove the node named by "tail". Please handle special cases in your method.




```
typedef struct node{
          int Data;
      struct node *Next;
} nodeLL;

nodeLL *head, *tail;

void removeTail()
{
```

```
}
```

2. (Arrays) ***(15 points)***

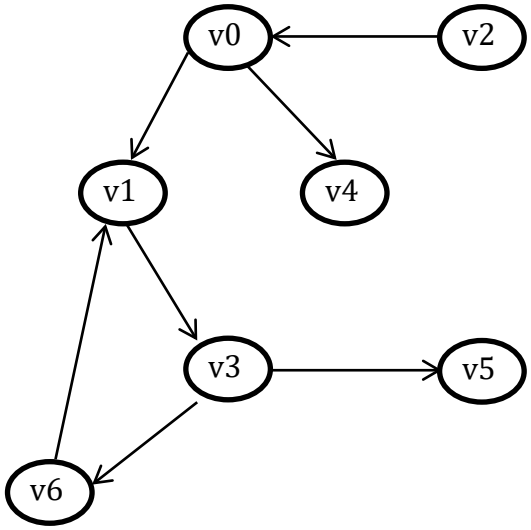
```
#include <stdio.h>
#define ROW 3
#define COL 4

void change(int array[][COL])
{
    int i, l, total;
    for(i = 0; i < ROW; i++){
        total = 0;
        for(l = 0; l < COL; l++){
            if(i == l)
                continue;
            total += array[i][l];
        }
        array[i][i] = total;
    }
}

   int main(void)
{
    int array[ROW][COL] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};
    int i, l;
    change(array);
    for(i = 0; i < ROW; i++){
        for(l = 0; l < COL; l++)
            printf("%d ", array[i][l]);
        printf("\n", array[i][l]);
    }
             return 0;
}
```

a. What does the ***change*** function do? Please explain.

b. Please re-write the same function **by using only point arithmetic**? The new function called ***change2*** is given below. Please fill out the blank boxes in the code. Let's assume that this function is called as ***change2(&dizi[0][0]).***

```
void change2(_____________)
{
    int *temp  = _________;
    int i, l, total;
    for(i = 0; i < ROW; i++){
        total = 0;
        for(l = 0; l < COL; l++, _______){
            if(i == l)
                continue;
            total += _________;
        }
        _______________ = total;
    }
}
```

3. (Graphs) ***(10 points)*** Consider this directed graph:

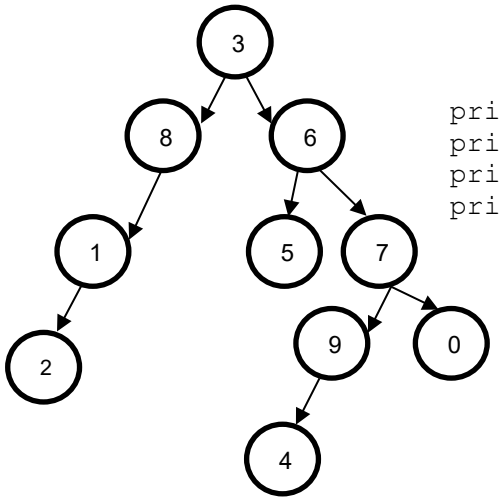


a. In what order are the vertices visited for a depth-first search that starts at v0?

b. In what order are the vertices visited for a breadth-first search that starts at v0?

Note : If a node v has more than one adjacent nodes pointing from v, please assume that they are going to be processed in increasing order according to their numbers. For example, for the adjacent nodes pointing from v3, firstly v5 is going to be processed, then v6.

4. (Trees) ***(12 points)*** Write a recursive method that prints the level of a given node in a binary tree. If the node is not found, it does not print anything. Assume all numbers are unique in the tree. Iterative solutions will not be accepted.




```
printLevel(root, 8, 0); // It prints "2"
printLevel(root, 4, 0); // It prints "5"
printLevel(root, 9, 0); // It prints "4"
printLevel(root, 12, 0);//It prints nothing
```

```
struct node
{
  int value;
  struct node *left;
  struct node *right;
};


void printLevel(struct node* root, int number, int level){









}
```

5. (Recursion and Performance Analysis) ***(12 points)*** According to the below recursive method, answer the following questions.

```
int recurse(int n)
{
    if (n <= 0)
        return 1;
    else
        return 1 + recurse(n/2);
}
```

a) What does the method return for n=8?

b) How many times will your method be called recursively for n=256?

c) What is the algorithm complexity of the recursive method in big Oh notation?

d) What is the total variable space needed for the execution of the method for an input size of 256, if the integer size is 4 bytes?

6. (Stack & Queues) ***(20 points)*** Complete the linked list stack implementation below for spaces "_____" as needed and inside the functions, and give the output of the code at the end.

```
typedef struct Node{
  struct ________ next;
  int data;
} Node;

typedef struct stackT{
  _____________ top;
  int maxSize; //max number of items allowed in the stack
  int count;  // current number of items in the stack
} stackT;

int StackIsEmpty(__________mystack)
{



}
int StackIsFull(__________mystack)
{




}
void StackPush( _________ mystack, int element)
{
  if (StackIsFull(________)) {
    printf("Can't push element on stack: stack is full.\n");
    exit(1);  /* Exit, returning error code. */
  }
  /* FILL BELOW, Put information in stack; update top and count. */









}
int StackPop(___________mystack)
{
  if (StackIsEmpty(________)) {
    printf(" Can't pop element stack is empty.\n");
    exit(1);  /* Exit, returning error code. */
  }
  /* update top and count, remove node from stack; return data of the node */









  return _________;
}
```

CONTINUE NEXT PAGE→

```
int main()
{
      stackT mystack;    /* A stack to hold ints. */
      mystack.maxSize=10;
      mystack.count=0;
      mystack.top=NULL;

      StackPush(____mystack, 5);
      StackPush(____mystack, 1);
      StackPush(____mystack, 7);

      printf(StackPop(____mystack));
      printf(StackPop(____mystack));
      printf(StackPop(____mystack));
      printf(StackPop(____mystack));

      return 1;
}
```

//Output of the above code : _____

7. a-(Tries) ***(8 points)*** Construct a trie from given key-value pairs:

cell	1
cope	5
call	7
caller	9
seller	8
sell	2
set	4
cop	3

b- (**Data Structures**) ***(7 points)*** Which data structure is the most appropriate for the following problems? No explanation is needed. (Possible answers: stack, queue, array, multidimensional array, linked list, circular linked list, sparse matrix, tree, trie, graph, hash table),

(a) When a printer receives several printing requests, _____

(b) Map of highway system used to display traffic travel times on a web page. The map displays principle cities, intersections, and major landmarks, the roads that connect them, and the travel times between them along those roads.

_____

(c) Chess board – an 8 x 8 board used for a game of chess. Each square on the board is either empty or contains a chess piece. _____

(d) When you want to store a collection of items in a set where the size is known.____

_____

(e) A list of the legal words in a Scrabble game. We want to be able to quickly check whether words used by players do, in fact, exist in the list. _____

(f) When you want to build a scheduler for processes where each process takes CPU cycles in rounds again and again, _____

(g) When it is checked by the computer if a list of parentheses is balanced,

_____