Matriculation Number: H T 0 4 0 8 3 1 R **CS1102C**

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
SEMESTER II AY2006/2007

CS1102C: DATA STRUCTURES AND ALGORITHMS

Mid Term Test Time Allowed: 1.5 Hours

MATRICULATION NUMBER:

H	T	0	4	0	8	3	1	R

INSTRUCTIONS TO CANDIDATES:

1. Write your matriculation number in the space provided above. Also write your matriculation number at the top of each sheet in the test paper. Shade your matriculation number on the OCR form. Remember to sign on the form.

2. This examination paper consists of **THREE (3) questions. Question 1 consists of 10 MCQ questions** and comprises **TWELVE (12)** printed pages including this front page.

3. Answer the **MCQ** questions by shading the **OCR** form and answer all of the other questions directly in the space given after each question. If necessary, use the back of the page.

4. Marks allocated to each question are indicated. Total marks for the paper is **100**.

5. This is a closed book examination and you can write in pencil.

EXAMINER'S USE ONLY				
Questions	**Possible**	**Marks**	**Grader**	**Check**
MCQ 1-10	40			
Q2	30			
Q3	30			
Total	100			

b, d, c, d, b
e, e, c, b, d

Question 1 (MCQ – 40 marks)

1) _____ describes the ability of a class to derive properties from a previously defined class.
   a) Encapsulation
   b) Inheritance
   c) Polymorphism
   d) Information hiding
   e) None of the above

2) A program is to be written that prints an invoice for a small store. A copy of the invoice will be given to the customer and will display

   - A list of items purchased
   - The quantity, unit price, and total price for each item.
   - The amount due.

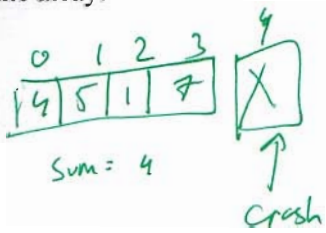
   Three candidate classes for this program are ***Invoice***, ***Item***, and ***ItemList***, where an *Item* is a single item purchased and *ItemList* is the list of all items purchased. Which class is a reasonable choice to be responsible for the ***amountDue*** function, which returns the amount the customer must pay?

   I) *Item* II) *ItemList* III) *Invoice*

   a) I only
   b) III only
   c) I and II only
   d) II and III only
   e) I, II and III

3) Refer to the following code segment. You may assume that *arr* is an array of integers and *arrSize* is the number of integers in the array.

```
int sum = arr[0],  i = 0;
while ( i < arrSize) {
    i++;
    sum += arr[i];
}
```



   Which of the following will be the result of executing the segment?

   a) Sum of arr[0], arr[1], …. , arr[arrSize – 1] will be stored in sum.
   b) Sum of arr[1], arr[2], …. , arr[arrSize – 1] will be stored in sum.
   c) The segment of codes accesses an array element that was not created by the program.
   d) An infinite loop will occur.
   e) None of the above

4) Refer to the function search:

```
// Return reference to first occurrence of key in list.
// Return NULL if key not in list.
// Precondition: node points to first node in list

ListNode * search (ListNode * node, int key) {

    < code>
}
```

Which of the following replacement for <code> will result in function search working as intended?

I)
```
if (node → getValue() = key)
    return node;
else
    return search(node → getNext(), key);
```

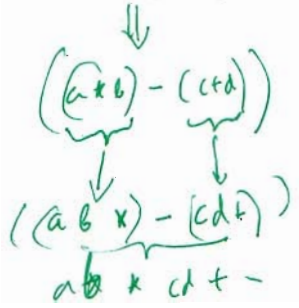
II)
```
ListNode * current = node;
while (current != null) {
    if (current → getValue() = key)
        return current;
    else
        current = current → getNext();
}
return null;
```

III)
```
ListNode * current = node;
while (current != null && !(current → getValue() = key))
    current = current → getNext();
return current;
```

a) I only
b) II only
c) III only
d) II and III only
e) I and II only

5) Which of the following is the postfix form of the infix expression: a * b – (c + d)?

a) a b c d + – *
b) a b * c d + –
c) a b c – * d +
d) a b c – d + *
e) a b c d - + *



6) Suppose that a queue q contains the value 10, 30, 20, 60, 50, 40 in that order, with 10 at the front of q. Suppose that there are just three operations that can be performed using only one stack, s.

I) Dequeue x from q then print x
II) Dequeue x from q then s.push(x);
III) Pop x from s then print x.

Which of the following is not a possible output using just these operations?

a) 10, 30, 20, 60, 50, 40
b) 20, 30, 10, 40, 50, 60
c) 60, 50, 40, 20, 30, 10
d) 30, 10, 60, 20, 40, 50
e) 10, 50, 40, 30, 20, 60

7) Suppose that ***s*** and ***t*** are both stacks of integers and ***s*** initially contains n integers, where n is large, and that ***t*** is initially empty. Assume further that length(***s***) gives the number of integers in ***s***. Which is true after execution of the following code segment?

```
int len = length(s) – 2;
int x;
for (int i = 0; i < len; i++) {
    x = s.top();      // top() returns the top element of stack
    s.pop();          // pop() pops the top element from stack
    t.push(x)         // push(x) pushes x onto stack
}
len = length(s) – 2;
for (int j = 0; j < len; j++) {
    x = t.top();
    t.pop();
    s.push(x);
}
```

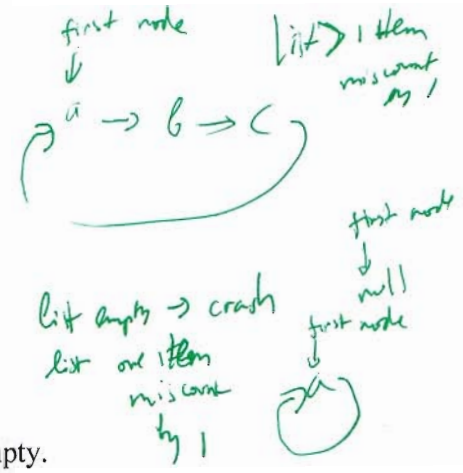
a) ***s*** is unchanged, and ***x*** equals the third item from the bottom of ***s***.
b) ***s*** is unchanged, and ***x*** equals ***s***.top().
c) ***s*** contains two elements, and ***x*** equals ***s***.top();
d) ***s*** contains two elements, and ***x*** equals the bottom elements of ***s***.
e) ***s*** contains two elements, and ***x*** equals ***t***.top().

8) How many base cases does a recursive binary search of a sorted array have?

a) 0
b) 1
c) 2
d) 3
e) 4

9) A circular linked list has a reference ***firstNode*** that points to the first element in the list, and is null if the list is empty. The following segment is intended to count the number of nodes in the list:

```
int count = 0;
ListNode * p = firstNode → getNext();
while (p != firstNode) {
    count++;
    p = p → getNext();
}
```

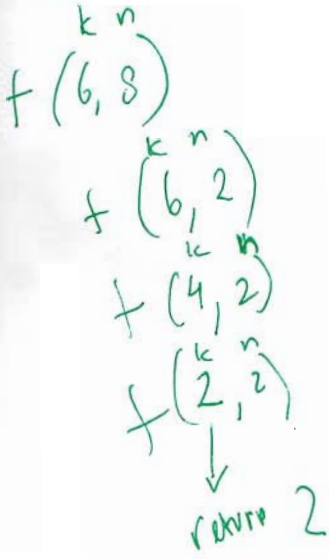


Which statement is true?

a) The segment works as intended in all cases.
b) The segment fails in all cases.
c) The segment works as intended whenever the list is nonempty.
d) The segment works as intended when the list has just one element.
e) The segment works as intended only when the list is empty.

10) Refer to the following recursive function:

```
int f(int k, int n) {
    if (n == k)
        return k;
    else
        if (n > k)
            return f(k, n – k);
        else
            return f(k-n, n);
}
```



What value is returned by the call f(6, 8)?

a) 8
b) 4
c) 3
d) 2
e) 1

Question 2 (30 marks)

a) The following table is used to represent a maze.

1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	1	2	2	2	1
1	1	1	2	1	1	2	1	1	9
1	2	1	2	2	1	2	1	2	1
1	2	1	1	2	1	2	2	2	1
1	2	1	2	2	1	1	1	2	2
2	2	1	1	1	2	1	1	1	1
0	1	1	2	2	1	1	1	1	1

Where 0: the start point; 9: the end point; 1: free space; 2: wall

Write a function which takes in a 2-d array with the maze configuration, no_of_row, no_of_column, x1, y1, (the start point) x2, y2 (the end point) as input and finds a path from the start point to the end point. The path should be indicated using the integer 3 as shown below. **(25 marks)**

3	3	3	3	3	3	3	3	3	3
3	2	2	2	2	1	2	2	2	3
3	3	3	2	1	1	2	1	1	9
1	2	3	2	2	1	2	1	2	1
1	2	3	3	2	1	2	2	2	1
1	2	3	2	2	1	1	1	2	2
2	2	3	1	1	2	1	1	1	1
0	3	3	2	2	1	1	1	1	1

Hints: 1. You may use a stack to solve this problem. For each location, you have to push its four neighbours if they are not visited before and they are not part of a wall. When you encounter a dead end, you pop from the stack and explore another path.

2. You may solve the problem recursively. Again, for each location, you have to make 4 recursive calls if it is a possible path to follow and the previous call did not return a successful path.

3. You may change some of the 1s in the table to other number (say 4) to indicate that you have gone through that location so that you don't try to visit it again.

```
bool done = false;

Void traverse (int [][] maze, int r, int c, int x, int y, int x_tgt, int y_tgt){
    if (done)
        return;
    if (x<0 || x>=c || y<0 || y>=r || maze[x][y] == 2 || maze[x][y] == 3)
        return;
    if (x == x_tgt && y == y_tgt) {
        maze[x][y] = 3; //flag this
        done = true;
        return;
    }
    maze[x][y] = 3; // flag this as potential path
    traverse(maze, r, c, x+1, y, x_tgt, y_tgt);
    traverse(maze, r, c, x, y+1, x_tgt, y_tgt);
    traverse(maze, r, c, x-1, y, x_tgt, y_tgt);
    traverse(maze, r, c, x, y-1, x_tgt, y_tgt);

    maze[x][y] = 4; // I have searched all 4 directions but this path don't lead to solution
}
```

allow the base case)

b) Assume that your function in part a) has successfully found the path and returned its result. Write a recursive function to output the path taken from the start point to the end point. For example, the path for the following maze is

(7, 0) (7, 1) (7, 2) (6, 2) (5,2) (4, 2) (3, 2) (2, 2) (2, 1) (2, 0) (1, 0) (0, 0) (0, 1) (0, 2) (0, 3) (0, 4) (0, 5) (0, 6) (0, 7) (0, 8) (0, 9) (1, 9) (2, 9)

(5 marks)

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	3	3	3	3	3	3	3	3	3	3
1	3	2	2	2	2	1	2	2	2	3
2	3	3	3	2	1	1	2	1	1	9
3	1	2	3	2	2	1	2	1	2	1
4	1	2	3	1	2	1	2	2	2	1
5	1	2	3	2	2	1	1	1	2	2
6	2	2	3	1	1	2	1	1	1	1
7	0	3	3	2	2	1	1	1	1	1

```
void print(int[][] maze, int r, int c, int x, int y){
if(x<0 || x>c || y<0 || y>R || maze[x][y]==1 or 2
refine
cout << "(" << x << ", " << y << ")";
maze[x][y]= 4;
print(maze, r, c, x+1, y)
                  x   , y+1
                  x-1 , y
                  x   , y-1
}
```

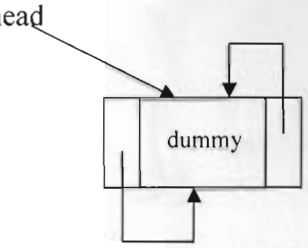
Question 3 (30 marks)

Assume that we have the following definition:

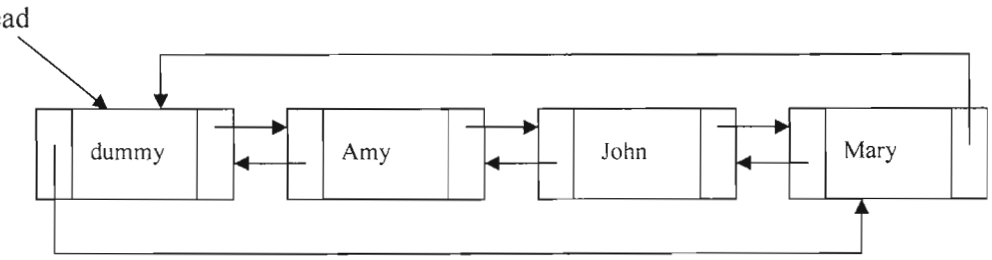
```
class Node {
    protected:
        struct ListNode {
            string name;
            ListNode * prev;
            ListNode * next;
        };
        typedef ListNode *NodePtr;
};
```

Consider the sorted doubly linked list shown below. This list is circular and has a dummy head node.

When it is empty



When it is not empty



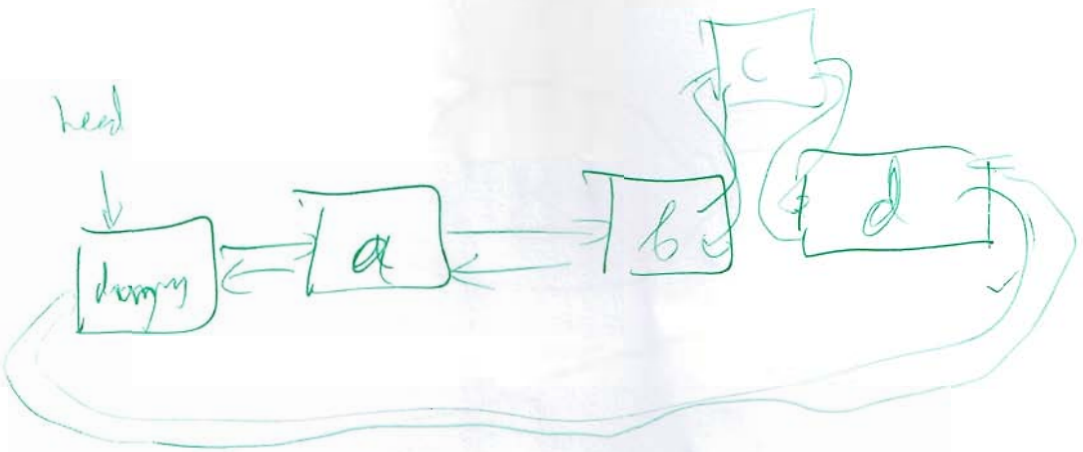
For the two parts in this question, you only need to write the required statements and not full functions. You may declare as many variables as you need.

assume if you are not found

a) Suppose that newName contains a name that you want to add to this list. Write some C++ statements that inserts a new node containing newName into its proper sorted order within the list. **(15 marks)**

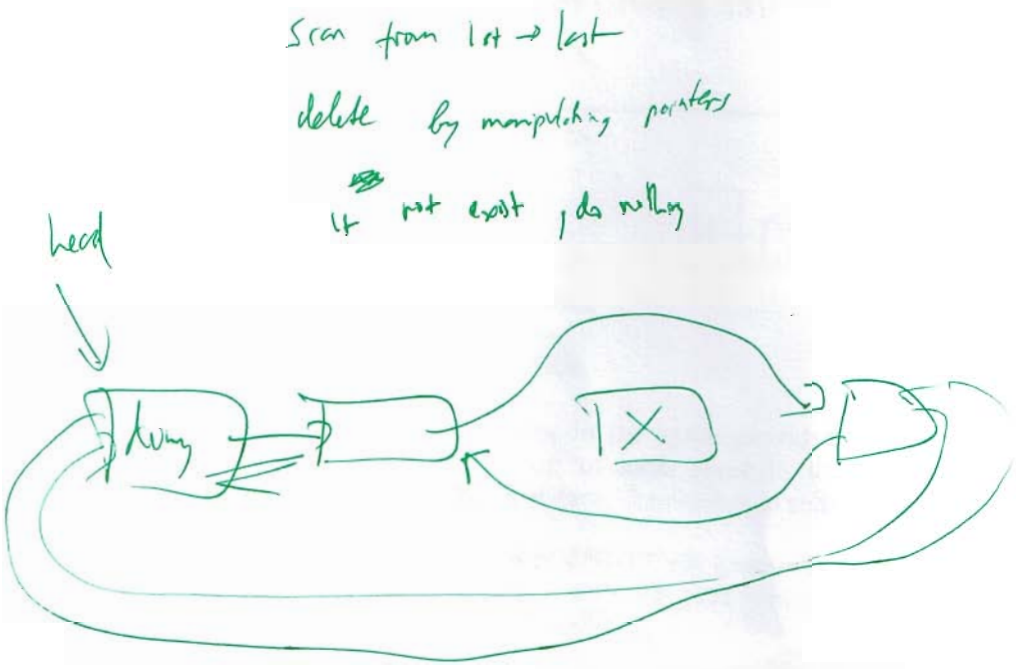
find the pos

insert by manipulating 4 pointers



b) Suppose that newName contains a name that you want to delete from the list. Write some C++ statements that delete the node containing newName.

(15 marks)



END-OF-PAPER