Distance Education Centre Victoria
Flexible learning for all

2015

NAME:

DECV Student ID:

ALGORITHMICS UNIT 3 & 4

Trial Exam 2: 2015 DECV

Reading Time: 15 minutes
Writing time: 120 minutes (2 hours)

QUESTION AND ANSWER BOOK

Section	*Number of questions*	*Number of questions to be answered*	*Number of marks*
A	20	20	20
B	9	9	80

- Students are permitted to bring into the examination room: pens, pencils, highlighters, erasers, sharpeners, rulers and one scientific calculator.
- Students are NOT permitted to bring into the examination room: blank sheets of paper and/or correction fluid/tape

Materials supplied

- Question and answer book of ?? pages
- Answer sheet for multiple-choice questions

Instructions

- Write your student number in the space provided above on this page.
- Check that your name and student number as printed on your answer sheet for multiple-choice questions are correct, and sign you name in the space provided to verify this.

- All written responses must be in English, point form is preferred.

Students are NOT permitted to bring mobile phones and/or any other unauthorised electronic devices into the test room.

SECTION A – Multiple Choice – select one option only

Question 1

The algorithmic design pattern that explores every possible decision in a deterministic way can be described as:

A. Divide and Conquer
B. Backtracking
C. Dynamic Programming
D. Brute Force

Question 2

The Queue is an Abstract data type with the following signature:

$name\ queue;$
$import\ elem, boolean;$
$ops \quad newQueue \ : \rightarrow queue;$
$enqueue : queue \times elem \rightarrow queue;$
$front \ : queue \rightarrow elem;$
$dequeue : queue \rightarrow queue;$
$isEmpty \ : queue \rightarrow boolean;$

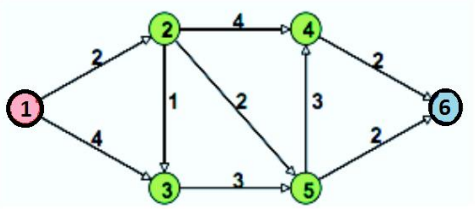
Given the variable MyQ is of type Queue, what does MyQ contain after the following operations, described in the pseudocode below.

```
newQueue(MyQ)
enqueue(MyQ,"Betty")
enqueue(MyQ,"Wilma")
enqueue(MyQ,"Fred")
if (front(MyQ) is equal to "Betty") then
      dequeue(MyQ)
      enqueue(MyQ,"Dino")
else
      enqueue(MyQ,"Barney")
End if
```

A. Betty, Wilma, Fred, Dino, Barney
B. Wilma, Fred, Dino
C. Wilma, Fred, Dino, Barney
D. Dino, Fred, Wilma, Betty

Question 3

Consider the following weighted directed digraph, G=(V,E).



Starting at node 1, the shortest path distances found using Dijkstra's algorithm to nodes 5 and 6 are:
A. 4 and 8 respectively
B. both are 6
C. 4 and 6 respectively
D. 4 and 7 respectively

Question 4

Dijkstra's algorithm finds the shortest path distances between nodes in graphs G=(V,E). Consider the following pseudocode from Wikipedia.

```
function Dijkstra(Graph, source):
    dist[source] ← 0
    prev[source] ← undefined
    create vertex set Q
    for each vertex v in Graph do
                if v ≠ source then                  // Initialise
                    dist[v] ← INFINITY
                    prev[v] ← UNDEFINED
                    add v to Q
                end if
    end do
    while Q is not empty do              // Find shortest path
       u ← vertex in Q with min dist[u]
       remove u from Q
       for each neighbor v of u do
          if (dist[u] + length(u, v))< dist[v] then
             dist[v] ← dist[u] + length(u, v)
             prev[v] ← u
          end if
       end do
    end do
    return dist[], prev[]
 end function
```

What is the worst case time-complexity for the execution of this version of Dijkstra's Algorithm?

A. O(|V|+|E|)
B. O(|V||E|)
C. O(log(|V|+|E|)
D. O($|V|^2$)

Question 5

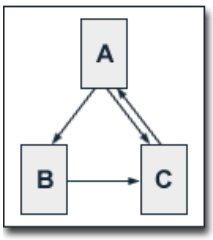
Describe the approach used in the pseudocode to describe an algorithm that reads text and checks if the words are palindromes using the block "checkPalindrome".

```
Algorithm ProcessText(Input List Words)
       newStack (PalStack)
       For each word in words do
              If (checkPalindrome(word) is true) then
                     Push(PalStack,word)
              End if
       End do
       While (PalStack is not empty) do
              say(top(PalStack))
              pop(PalStack)
       end do
end Algorithm
```

A. Backtracking
B. Recursion
C. Modularisation
D. Dynamic Programming

Question 6

The Page ranking where the damping constant p=0.85 for the following web site can be calculated using the probability recurrences:



A.
$$PR(A) = \frac{1-0.85}{3} + 0.85(PR(C))$$
$$PR(B) = \frac{1-0.85}{3} + 0.85\,(PR(A)\,/\,2)$$
$$PR(C) = \frac{1-0.85}{3} + 0.85\,(PR(A)\,/\,2 + PR(B))$$

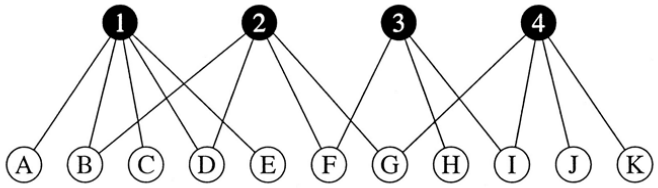
B.
$$PR(A) = 0.15 + 0.15\,(PR(C))$$
$$PR(B) = 0.15 + 0.15\,(PR(A)\,/\,2)$$
$$PR(C) = 0.15 + 0.15\,(PR(A)\,/\,2 + PR(B))$$

C.
$$PR(A) = 0.15 + 0.85\,(PR(C))$$
$$PR(B) = 0.15 + 0.85\,(PR(A)\,/\,2)$$
$$PR(C) = 0.15 + 0.85\,(PR(A)\,/\,2 + PR(B))$$

D.
$$PR(A) = \frac{1-0.85}{3} + 0.5\,(PR(C))$$
$$PR(B) = \frac{1-0.85}{3} + 0.5\,(PR(A)\,/\,2)$$
$$PR(C) = \frac{1-0.85}{3} + 0.5\,(PR(A)\,/\,2 + PR(B))$$

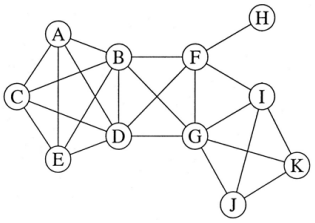
Question 7

The tree graph below shows 4 different companies numbered 1, 2, 3, 4 as root nodes. Each company has a board consisting of members as shown. Some members can be on the board of multiple companies. For Example person B is on the board of company 1 and 2.

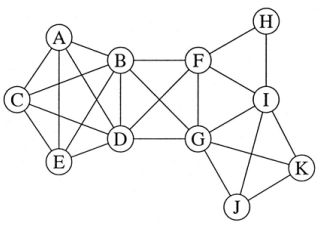


An alternate way of representing this same information is by the following graph:

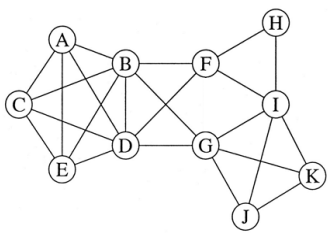
A.



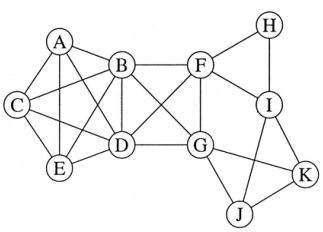
B.



C.

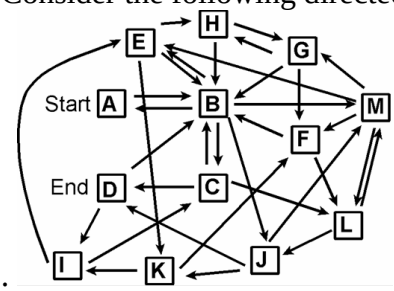


D.



Question 8

Consider the following directed graph



When finding the Hamiltonian path in a directed graph; which of the following statements is **not** true.

A. The problem is NP-Hard.
B. The solution can be checked in exponential time.
C. The solution can be checked in polynomial time.
D. The total number of paths grows exponentially as the number of nodes increases.

Question 9

Consider the following pseudocode for traversing a tree to find a leaf node that meets a goal criteria:

```
function seek(Input Node)
    if Node is a leaf node then
        if (leaf is a goal node) then
            return true
        else
            return false
        end if
    else
        for each child of node do
            if seek(child) succeeds then
                return true
        end do
        return false
    end if
end function
```

The description that best fits the design pattern of the algorithm above is:

A. Backtracking
B. Divide and Conquer
C. Dynamic Programming
D. Greedy

Question 10

Consider the time complexity described by the following recurrence:

$$T(n) = 3T\left(\frac{n}{2}\right) + n^2$$

Expressed a function of n the equivalent time complexity is given by:

A. $T(n) = O(n^2)$
B. $T(n) = O(n^2 logn)$
C. $T(n) = O(nlogn)$
D. $T(n) = O(n^{\log_2 3})$

Question 11

Consider the following pseudocode:

```
Function Mystery(n)
      If (n<=1) then
            Return n
      Else
            Return (Mystery(n-1))*2
      End if
End function
```

What is the recurrence relation for the time complexity of Mystery?

A. T(n)=T(n-1)+O(n)
B. T(n)=2T(n-1)+O(n)
C. T(n)=T(n-1)+T(n-2)
D. T(n)=T(n-1)+O(1)

Question 12

Consider the following block coded in edgy:

If the recursive block hello is called  then the expected result is:

A. 122.210
B. 112.211
C. 111.111
D. 101.101

Question 13

The worst case time complexity for Depth First search and Breadth First Search Traversal of a Graph G=(V,E) is respectively equal to:

A. $O(|V||E|), O(|V|^2)$
B. Both are $O(|V|+|E|)$
C. Both are $O(|V|\log|E|)$
D. $O(|V|^2), O(|V||E|)$

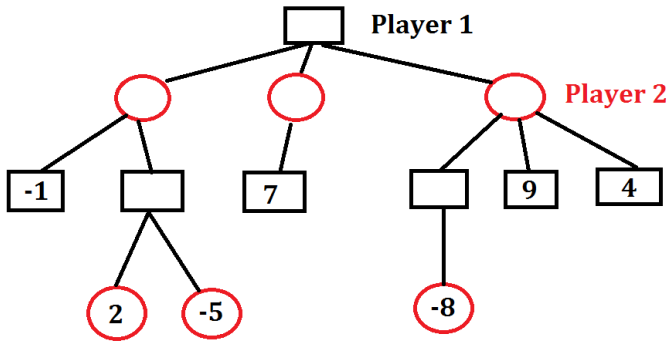
Question 14

One of the main drawbacks of hill-climbing heuristic approach for solving NP-Complete problems is:

A. Terminates at local optimum
B. Terminates at global optimum
C. Views the solution space as a landscape
D. Fails to find a good solution

Question 15

Consider the following game tree for Player 1 and Player 2 with leaves showing the benefit to Player 1.



If the Minimax algorithm is run on this tree then the root node with have the value of:

A. 8
B. -1
C. 7
D. 9

Question 16

Which problem is the odd one out in the list below?

A. Coin Change Problem
B. Travelling Salesman Problem
C. Knapsack Problem
D. Minimum Graph colouring problem

Question 17

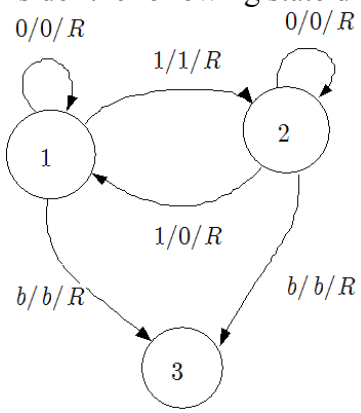
Which option best completes the missing words for the following paragraph?

DNA Computing performs computations using _____, the benefits of DNA computing include massive _____ processing of polynomial problems. DNA Computing offers _____ in the solution of intractable problems.

A. chemical reactions, serial, improvement
B. chemical molecules, serial, no improvement
C. biological processes, parallel, feasibility
D. chemical processes, parallel, no new capability

Question 18

Consider the following state diagram for a Turing Machine, where "b" indicates a blank:



If this 3 state Turing machine is run on the tape containing the sequence
011101 and is starting in state 1 then the machine will halt with the sequence and state of:

A. 010100, state 2
B. 010011, state 2
C. 010100, state 1
D. 010010, state 1

Question 19

The Church-Turing thesis concerns the notion of an *algorithm*, M, for achieving some desired result, which of the following statements is **not** true about the Church-Turing thesis regarding algorithm M?

A. Algorithm M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
B. Algorithm M will, if carried out without error, produce the desired result in a finite number of steps;
C. Algorithm M can (in practice or in principle) be carried out by a human being unaided by any machinery save paper and pencil;
D. Algorithm M demands insight and ingenuity on the part of the human being carrying it out.

Question 20

The Church-Turing thesis, the Turing Machine and Lambda calculus are part of the history of Computer Science. They evolved from ideas emanating from Hilbert's program of the formalism of mathematics. Which of the following statements is **not** true regarding Hilbert's program of formalisation?

A. The aim is to enumerate all the symbols used in mathematics and logic.
B. Formalism attempts to characterize unambiguously all the combinations of symbols which represent statements classified as 'meaningful' in classical mathematics.
C. Hilbert's intention is to supply a construction procedure which enables us to construct successively all the formulas which correspond to the 'provable' statements of classical mathematics.
D. It is possible to achieve the formalisation of all of mathematics in axiomatic form, together with a proof that this formalisation of mathematics is consistent.

SECTION B – Extended Response Questions Answer all questions in the space provided.

In questions where more than one mark is available, appropriate working **must** be shown.

Question 1 (6 marks)

Given a 2-litre jug, and a 1-litre jug; neither have any measuring marks on them at all. Initially both are empty, therefore the initial state can be represented by **0-0**. You need to get exactly one litre into the 2-litre jug and the 1-litre jug using unlimited tap water to get to the target state of **1-1**.

a. Create a transition graph showing the possible states of the solution to the jug problem together with the legal transition information on each edge. (3 marks)

b. List all the sequences of states from initial to target that form a solution to this problem. (1 mark)

c. How can you best find the solution with the smallest number of transitions? Explain. (2 marks)

Question 2 (10 marks)

Consider the following problem where there are a row of "n" coins laid out on a table whose values are some positive integers c1, c2, c3,…….cn, which are not necessarily distinct.

The goal is to pick up the maximum amount of money subject to the rule that no two neighbouring coins in the initial row can be selected.

M(n) is the function giving maximum amount that can be picked from a row of n coins, when they have been partitioned into two groups. The first includes the last coin (cn) and the second does not.

- M(n)=max[cn + M(n-2),M(n-1)] for n>1
- M(1)=c1 if there is one coin
- M(0)=0 if there are no coins

Here is a naïve algorithm for solving this problem:

```
Function MaxC(Input a list of coins: coinList)
      if (length(coinList)== 0) then
            return 0
      else if (length(coinList)==1) then
            return coinList[1]
      else
            lastcoin:=length(coinList)
            choice1:= coinList[lastcoin] + MaxC(coinList[1:lastcoin – 2])
            choice2:= MaxC(coinList[1:lastcoin-1])
            return maximum(choice1, choice2)
      end if
end function
```

a. If the coins laid out have the denominations in sequence {5,1,2,8}, show the call tree for MaxC({5,1,2,8}) for the naïve algorithm together with the input list at each call for this instance of the problem. (2 marks)

b. Give the recurrence relation describing the time complexity in the naïve algorithm. (1 marks)

Question 2 (continued)

c. What design pattern could be used to improve this naïve algorithm? What kind of benefits could be achieved by selecting a different design pattern to improve the time complexity of this algorithm? Justify your answer. (2 marks)

d. Write your new algorithm in the design pattern you have chosen in d. (3 marks)

e. Compare the time complexity for the naïve and your improved version algorithms. (2 marks)

Question 3 (4 marks)

Give the count of the number of times that the command "S" is executed for the following code fragments.

a. Explain your reasoning. (2 marks)

```
I:=1
While I ≤ (N − 1) do
      J:=1
      While J ≤ (N − 1) do
            S
            J:=J+1
      End do
      I:=I+1
End do
```

b. Explain your reasoning. (2 marks)

```
I:=1
While I ≤ N do
   J:=1
   While J ≤ N do
         S
         J:=J+1
   End do
   K:=1
   While K ≤ N do
         L:=1
         While L ≤ N do
               S
               L:=L+1
         End do
         K:=K+1
   I:=I+1
End do
```

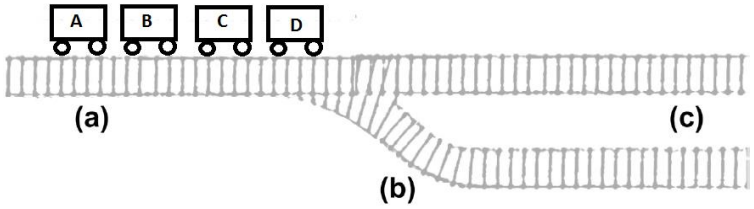
Question 4 (8 marks)

A large number of people enter the Thornbury Coles at the same time, resulting in a line of 8 people waiting to be served at one register (Register A).
This can be modeled with a queue: RegisterA = queue(P1; P2; P3;P4;P5;P6;P7;P8) where P1 is at the front of the queue.
The manager decides to open another register (Register B) and is shifting the back half of the line to the new register (Register B).

a. Write a sequence of formal Abstract Data Type operations such that the result is: (2 marks)

RegisterA = queue(P1; P2; P3; P4)
RegisterB = queue(P5; P6; P7; P8)

The railway line goes from (a) to (c) with a siding at (b) that can hold one or more train carriages.
Consider the following carriage configuration at point (a)

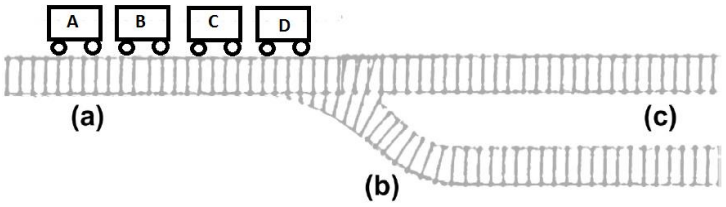


b.
(i) Write an algorithm in pseudocode to re-arrange the order that gets to point (c) to be DCBA using the siding (b). (2 marks)

Question 4 (continued)

(ii) Write a generalised algorithm to re-arrange any given order of carriages at point (a) to the reverse order at point (c) (2 marks)

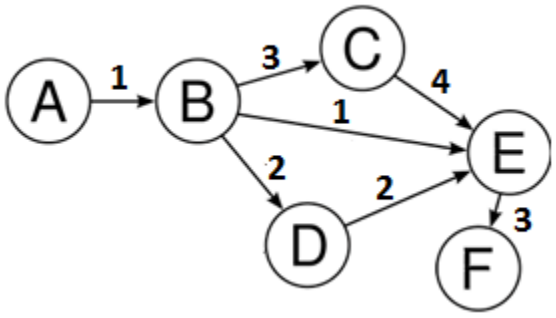
c. Consider the following carriage configuration at point (a)



Write an algorithm in pseudocode to re-arrange the order that gets to point (c} to be ACBD using the siding (b). (2 marks)

Question 5 (12 marks)

Consider the following directed graph:




```
function Dijkstra(Graph, source):
    dist[source] ← 0
    prev[source] ← undefined
    create vertex set Q
    for each vertex v in Graph do
                if v ≠ source then                    // Initialise
                    dist[v] ← INFINITY
                    prev[v] ← UNDEFINED
                    add v to Q
                end if
    end do
    while Q is not empty do                 // Find shortest path
       u ← vertex in Q with min dist[u]
       remove u from Q
       for each neighbor v of u do
          if (dist[u] + length(u, v))< dist[v] then
             dist[v] ← dist[u] + length(u, v)
             prev[v] ← u
          end if
       end do
    end do
    return dist[], prev[]
end function
```

a. Using Dijkstra's Algorithm, update a set S for shortest paths from node A. (2 marks)

Iteration	S	Dist(A)	Dist(B)	Dist(C)	Dist(D)	Dist(E)	Dist(F)
0	{A}	0	1				
1	{A,B}						

b. (i) What design pattern is used by Dijkstra's algorithm? (1 marks)

_____

_____

(ii) Give an argument for the correctness of this algorithm? (2 marks)

_____

_____

_____

_____

c. What is the definition of Transitive closure with respect to a graph? (1 mark)

Recall that vertices u and v are adjacent in a graph G if there is an edge <u,v>. This information can be stored in a Boolean adjacency matrix of size |V| × |V| called A; each cell of A[i,j] defines adjacency between vertex from A[i] to vertex A[j] where 0 indicates no edge and 1 indicates an edge exists.

Consider the Algorithm for finding the Transitive Closure of a graph.

```
Input --Graph G = <V, E>
// A is the Adjacency Matrix, C is the Connectivity Matrix
// notation A[i,j] indicates row i, column j
for vertex k in 1..|V| do
   for vertex i in 1..|V| do
      for vertex j in 1..|V| do
          if (A[i,j]==1 or (A[i,k]==1 and A[k,j]==1)) then
                  C[i,j]:= 1
      end for
   end for
end for
```

the adjacency matrix **"A"** for the graph shown in part a.

A =

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	1	1	1	0
C	0	0	0	0	1	0
D	0	0	0	0	1	0
E	0	0	0	0	0	1
F	0	0	0	0	0	0

C =

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

d. Complete the connectivity matrix **"C" above** for the graph shown in part a. (2 marks)

e. Identify any loop invariants used by the algorithm above. (1 mark)

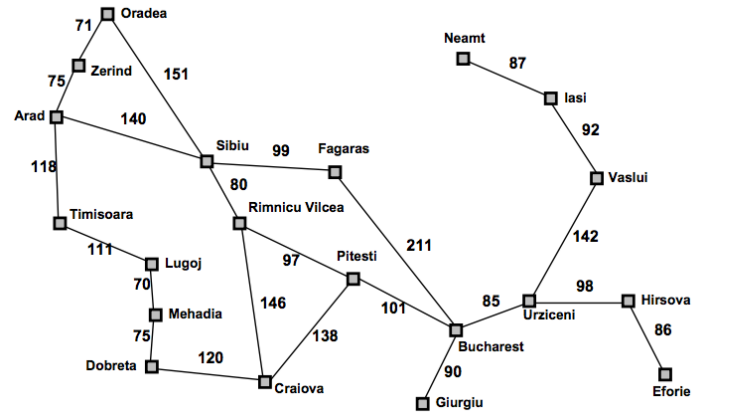
f. Make an argument for the correctness of the algorithm. (3 marks)

Question 6 – (12 Marks)

a. Describe the idea behind the hill climbing heuristic algorithm making reference to its origins as an optimisation methodology, describe any drawbacks that may occur using this heuristic. (3 marks)

b. What are the principles of the simulated annealing heuristic algorithm? How does it find a "good" solution for a problem. (2 marks)

c. Why are heuristics used in computer science to solve problems? Give an example of a problem that uses heuristic algorithms. What features of that problem indicate a heuristic solution needs to be used? (3 marks)



Consider the following road map of a small region of Romania.

Dorina who is a Computer Scientist lives in Oradea and wishes to visit her friend Anton who lives in Bucharest.

d. Describe a "Best First Search" approach that could be used by Dorina to find the best route from the town of "Oradea" to the city of "Bucharest". (2 marks)

e. What are the principles of the best first heuristic algorithm? How does it differ from Greedy Algorithm strategies? How does it find a "good" solution for a problem. (2 marks)

Question 7 (10 Marks)

a. What is the Turing Test? (2 marks)

b. Could we apply the Turing test to a computer that plays chess? Why or why not? (2 marks)

c. Describe the Searle's Chinese Room Experiment and its conclusions about Artificial Intelligence. (2 marks)

d. In John Searle's Chinese Room Experiment what are the standard replies to the main arguments in support of Artificial Intelligence: The Systems Argument; The Robot Argument; The Brain Simulator Argument; The Other Minds Argument ? (4 marks)

Question 8 (8 marks)
Consider the following signature description for the List Abstract Data Type:

$$
\begin{array}{l}
name\ list; \\
import\ elem, boolean; \\
ops \quad newList \quad : \rightarrow list; \\
\qquad first \quad : list \rightarrow elem; \\
\qquad rest \quad : list \rightarrow list; \\
\qquad insert \quad : list \times elem \rightarrow list; \\
\qquad contains \quad : list \times elem \rightarrow boolean; \\
\qquad append \quad : list \times elem \rightarrow list; \\
\qquad isEmpty \quad : list \rightarrow boolean;
\end{array}
$$

a. Using the signature descriptions for the list write a series of Pseudocode operations that removes all vowels from the contents of a given list of letters. You may use two lists one as input and one as output for this problem. (2 marks)

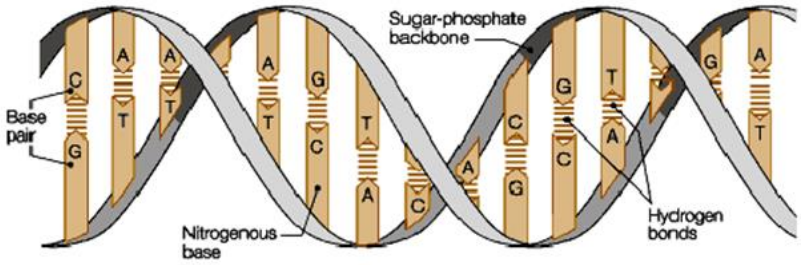
b. Complete the Pseudocode for the recursive algorithm RevWord to reverse a word. (2 marks)

```
Function RevWord(Input Word, Wordlength)
// Word is a list of letters
// Wordlength is the length of the list
   If Wordlength < 1 then
      [                                    ]
   Else
      [                                                    ]
   End if
End function
```

c. What is tail recursion? Is it possible to convert non-tail recursive algorithms to tail recursive ones? Describe how this is done? (2 marks)

d. Write a tail-recursive version in Pseudocode of the Reverse Word Algorithm. (2 marks)

Question 9 (10 marks)



a. Describe the benefits and drawbacks of DNA Computing for solving problems. (2 marks)

b. Can DNA Computing solve NP-Hard problems? Explain you reasoning. (2 marks)

In the 1990's Adelman used DNA Computing to solve an instance of the Travelling Salesman problem. Below are two tables containing some of the information about the cities and flights connecting them from his experiment.

CITY
ATLANTA
BOSTON
CHICAGO
DETROIT

FLIGHT
ATLANTA - BOSTON
ATLANTA - DETROIT
BOSTON - CHICAGO
BOSTON - DETROIT
BOSTON - ATLANTA
CHICAGO - DETROIT

c. What methods did Adleman use to encode the vertex and edge information for his city network? (2 marks)

d. What algorithm was used by Adelman to establish the optimum tour of this network? Give a brief outline of the algorithm's pseudocode. (2 marks)

e. Describe how the results were identified as a solution to this problem? (2 marks)

END OF TRIAL EXAM 2