codebash

OCR A Level Computer Science H446

H446/02 Algorithms and programming

Mock examination | Set A

Time allowed: 2 hours 30 minutes
You must have: a black pen, an HB pencil and a ruler
You must not use: a calculator

Candidate name _____

Class _____ **Candidate number** _____

Instructions

- Use black ink. You can use an HB pencil, but only for graphs and diagrams.
- Write your answer to each question in the space provided. If you need extra space, use the additional answer page at the end of this paper and write the question number(s) clearly.
- Answer **all** the questions.
- Where a question asks you to write code, you may answer in OCR pseudocode or a high-level programming language such as Python or C#.

Information

- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended response will be assessed in questions marked with an asterisk (*).
- This paper has two sections. Section A is worth 100 marks and Section B is worth 40 marks.
- This document has **25** pages.

For teacher use

Question	Max	Mark
1	19	
2	18	
3	16	
4	15	
5	12	
6	20	
7	40	
Total	**140**	

Student self-assessment (complete after the mock)	
What went well?	
Where can I improve, and how?	
Percentage:	**Grade:**

This is an independently produced practice paper. It is not produced, endorsed or checked by OCR.
© Decide Learning Ltd (CodeBash). Licensed for use within the downloading school only.

Section A

Answer **all** the questions.

1 BusLink is developing a journey planner for a city's bus network.

(a) The journey planner models the bus network as a graph. Each node is a bus stop and each edge is a road between two stops, weighted with the travel time.

Explain what is meant by abstraction. Identify **two** details of the real bus network that can be left out of the model, explaining why each can be left out.

[4]

(b) A function `findRoute(startStop, endStop, departTime)` returns the quickest route.

Identify **one** output of the function and **two** preconditions that must be true for the function to work correctly.

Output

Precondition 1

Precondition 2

[3]

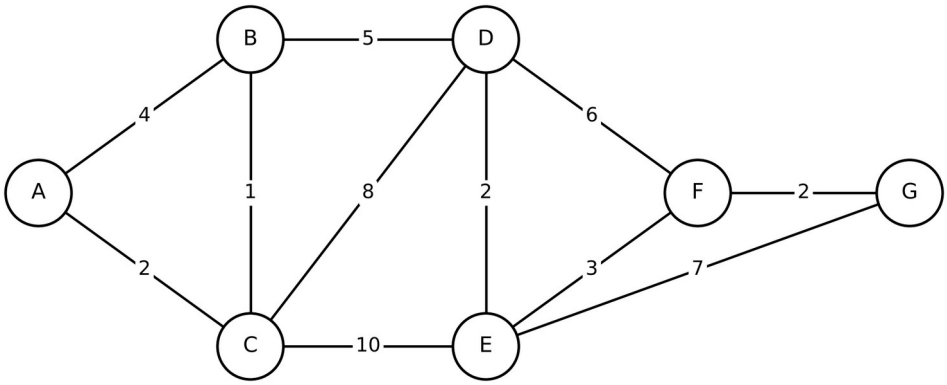
© Decide Learning Ltd (CodeBash)

(c) The journey planner caches the results of recent searches.

Describe what is meant by caching. Give **one** benefit and **one** drawback of caching for the journey planner.

[3]

(d) Part of the bus network is shown. The weights are travel times in minutes.



(i) Complete the table to show how Dijkstra's algorithm finds the quickest route from stop A to every other stop. Where a distance is replaced by a shorter one, cross out the old value and write the new value next to it.

Stop	Visited (order)	Quickest time from A	Previous stop
A		0	
B			
C			
D			
E			
F			
G			

[4]

© Decide Learning Ltd (CodeBash)

(ii) State the quickest route from A to G and its total time.

Route

Time

[2]

(e) BusLink is considering using the A* algorithm instead of Dijkstra's algorithm.

Describe how A* differs from Dijkstra's algorithm.

[3]

© Decide Learning Ltd (CodeBash)

2 The recursive function `power` returns x raised to the power n, where n is a whole number and n ≥ 0.

```
 1  function power(x, n)
 2      if n == 0 then
 3          return 1
 4      elseif n MOD 2 == 0 then
 5          half = power(x, n DIV 2)
 6          return half * half
 7      else
 8          return x * power(x, n - 1)
 9      endif
10  endfunction
```

(a) **(i)** State the line number where the base case is tested.

[1]

(ii) The function is called with `power(3, 5)`.

Complete the table to show each call of the function, in the order the calls are made, and the value each call returns. You may not need all the rows.

Call	Value returned

[4]

© Decide Learning Ltd (CodeBash)

(b) Rewrite the function `power` so that it uses iteration instead of recursion. It must return the same result for any whole number x and any n ≥ 0. Your function does not need to use the halving method.

[4]

(c)* A programmer says that recursive functions such as `power` should always be rewritten to use iteration.

Discuss this view. In your answer, refer to how recursion uses the stack, memory use, efficiency and how easy the code is to understand. Use the function `power` as an example.

© Decide Learning Ltd (CodeBash)

[9]

© Decide Learning Ltd (CodeBash)

3 **(a)** The list below is sorted into ascending order using an insertion sort.

`[27, 9, 41, 13, 5]`

Show the list after each pass of the insertion sort.

After pass 1

After pass 2

After pass 3

After pass 4

[3]

(b) Describe how a merge sort would sort this list into ascending order:

`[38, 16, 52, 7, 29, 44, 11, 23]`

You should show the contents of the sub-lists at each stage.

[4]

© Decide Learning Ltd (CodeBash)

(c) Complete the table to give the Big O complexity for each case.

Algorithm and case	Big O complexity
Insertion sort, best-case time	
Insertion sort, worst-case time	
Merge sort, worst-case time	
Merge sort, space	

[4]

(d) Write a function `binarySearch(items, target)` that uses a binary search to find target in the array items. The array is sorted into ascending order.

The function must return the index of target, or -1 if target is not in the array.

[5]

© Decide Learning Ltd (CodeBash)

4 A print service holds jobs in a circular queue. The queue is stored in an array `jobs` with six elements, indexed 0 to 5. The global variable `front` holds the index of the first job, `rear` holds the index of the last job and `size` holds the number of jobs in the queue.

The current state of the queue is shown. front = 4, rear = 1 and size = 4. dequeue() does not clear the element it removes.

Index	0	1	2	3	4	5
jobs	J3	J4			J1	J2

(a) The following operations are carried out in this order:

`dequeue()`, `enqueue("J5")`, `enqueue("J6")`

Show the values of front, rear and size, and the contents of the array, after these operations.

Index	0	1	2	3	4	5
jobs						

front

rear

size

[3]

© Decide Learning Ltd (CodeBash)

(b) Write the function `dequeue()`. It must return the job at the front of the queue and update the queue. If the queue is empty, it must return "EMPTY".

[5]

(c) These tree names are inserted, in this order, into an empty binary search tree. Names are compared alphabetically.

Maple, Cedar, Rowan, Birch, Elm, Oak, Willow, Ash

Draw the binary search tree. Make it clear which child of each node is on the left and which is on the right.

[3]

© Decide Learning Ltd (CodeBash)

(d) Give the order in which the nodes of your tree are visited by:

(i) a breadth-first traversal

[2]

(ii) a depth-first (post-order) traversal.

[2]

© Decide Learning Ltd (CodeBash)

5 A school is developing an app that creates staff and student timetables.

(a) Explain **one** benefit of using reusable program components when developing the app.

[2]

(b) The task "generate a timetable" is to be broken down into sub-procedures.

Identify **three** sub-procedures that could be used, and state the order in which they must run. Justify your order.

[3]

© Decide Learning Ltd (CodeBash)

(c) The app's login algorithm is shown.

```
 1  attempts = 0
 2  loggedIn = false
 3  do
 4      user = input("Username: ")
 5      pwd = input("Password: ")
 6      if checkDetails(user, pwd) then
 7          loggedIn = true
 8      else
 9          attempts = attempts + 1
10      endif
11  until loggedIn OR attempts == 3
12  if loggedIn then
13      print("Welcome")
14  else
15      print("Account locked")
16  endif
```

Identify **three** decision points in the algorithm. For each, state the condition checked.

1

2

3

[3]

© Decide Learning Ltd (CodeBash)

(d) The procedure `adjust` and the main program below are run. In the procedure, a is passed by value and b is passed by reference.

```
 1  procedure adjust(a:byVal, b:byRef)
 2      a = a * 2
 3      b = b + a
 4      print(str(a) + " " + str(b))
 5  endprocedure
 6
 7  x = 3
 8  y = 4
 9  adjust(x, y)
10  print(str(x) + " " + str(y))
11  adjust(y, x)
12  print(str(x) + " " + str(y))
```

Give the four lines of output.

1

2

3

4

[4]

© Decide Learning Ltd (CodeBash)

6 **(a)** A program finds a route through a maze using backtracking.

Describe how backtracking finds a route through the maze.

[3]

(b) Binary search is an example of divide and conquer.

Describe what is meant by divide and conquer.

[2]

© Decide Learning Ltd (CodeBash)

(c) The function below should return the mean of the values in the array marks, but it returns the wrong result.

```
1  function average(marks)
2      total = 0
3      for i = 1 to marks.length - 1
4          total = total + marks[i]
5      next i
6      return total / marks.length
7  endfunction
```

Describe how **two** features of an IDE could be used to find the error in the function.

[3]

(d)* FreshCart, a chain of supermarkets, wants to use computational methods to decide how much stock each shop should order and to plan its delivery lorry routes.

Discuss how FreshCart could use data mining, heuristics and performance modelling. In your answer, evaluate the benefits and limitations of each method for FreshCart.

© Decide Learning Ltd (CodeBash)

[12]

© Decide Learning Ltd (CodeBash)

Section B

Answer **all** the questions.

Section B scenario. ShelfBot Ltd writes software that controls robots in a warehouse.

The warehouse floor is a grid of 20 by 12 squares. Each square is identified by its x coordinate (0 to 19) and its y coordinate (0 to 11). Robots move between squares to collect items. Moving one square left, right, up or down uses 2% of the robot's battery. Robots cannot move diagonally.

Each robot is an object of the class Robot. The attribute battery is a whole number from 0 to 100 that shows the percentage of charge left.

The function `abs(n)` returns the absolute value of n. For example, abs(-4) returns 4.

Part of the class is shown.

```
 1  class Robot
 2      private id
 3      private x
 4      private y
 5      private battery
 6
 7      public procedure new(givenId, startX, startY)
 8          // to be written in part (b)
 9      endprocedure
10
11      public function getX()
12          return x
13      endfunction
14
15      public function getY()
16          return y
17      endfunction
18  endclass
```

7 **(a)** The program stores a plan of the warehouse floor in a two-dimensional array `floor[20, 12]`. Each element holds "EMPTY", "SHELF" or "CHARGER".

Explain why a two-dimensional array is a suitable data structure for the floor plan.

[3]

© Decide Learning Ltd (CodeBash)

(b) Write the constructor for the class Robot. It must set id and the starting position from its parameters and set battery to 100.

[4]

(c) Write the methods `getBattery()` and `setBattery(level)`. setBattery must only change battery if level is from 0 to 100 inclusive.

[3]

(d) Write program code for the main program to create a Robot object called r1 with id "R1" at square (0, 0), and then output r1's battery level.

[3]

© Decide Learning Ltd (CodeBash)

(e) The method `moveTo(newX, newY)` should move the robot to the square (newX, newY) if the square is on the grid and the robot has enough battery for the move (2% per square moved). It then reduces the battery by 2 for each square moved. The method contains **three** errors.

```
 1  public procedure moveTo(newX, newY)
 2      if newX < 0 OR newX > 20 OR newY < 0 OR newY > 11 then
 3          print("Invalid square")
 4      else
 5          steps = abs(newX - x) + abs(newY - y)
 6          if battery >= steps * 2 then
 7              x = newX
 8              y = newX
 9              battery = battery + steps * 2
10          else
11              print("Not enough battery")
12          endif
13      endif
14  endprocedure
```

Identify the line number of each error and write the corrected line.

Line number	Corrected line

[3]

© Decide Learning Ltd (CodeBash)

(f) The positions of the charging points are stored in a two-dimensional array `chargers`. For charger i, chargers[i, 0] holds its x coordinate and chargers[i, 1] holds its y coordinate. The number of chargers is passed as numChargers.

The number of steps from a robot to a charger is the difference in x plus the difference in y, ignoring signs.

Write a method `nearestCharger(chargers, numChargers)` for the class Robot. It must return the index of the charger that needs the fewest steps from the robot's current position. If two chargers need the same number of steps, it must return the lower index.

For example, a robot at (10, 5) with chargers at (0, 0), (19, 11), (12, 2) and (8, 8) needs 15, 15, 5 and 5 steps, so the method returns 2.

[6]

© Decide Learning Ltd (CodeBash)

(g) Explain why planning the routes of the robots is a problem that can be solved using computational methods.

[3]

(h) ShelfBot also uses flying drones. A class Drone inherits from Robot. Drones can fly over shelves, so their version of moveTo uses 3% of the battery per square.

(i) Describe how inheritance and polymorphism would be used in the class Drone and the main program.

[4]

(ii) Explain why x, y and battery are declared as private attributes.

[2]

(i)* Several robots work at the same time. Each robot is controlled by its own thread, and all the threads share a list of charging points that are free.

Discuss the benefits of concurrent processing for the warehouse system, the problems that could arise when the robots share data, and how the system could be designed to prevent these problems.

© Decide Learning Ltd (CodeBash)

[9]

© Decide Learning Ltd (CodeBash)

END OF QUESTION PAPER

Additional answer space. If you use this page, write the question number clearly in the margin.

© Decide Learning Ltd (CodeBash)