|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name:** |  | **Index Number:** |  | **Class:** |  |

|  |  |
| --- | --- |
| DHS logo CMYK | **DUNMAN HIGH SCHOOL**  **Preliminary Examination**  **Year 6** |

|  |  |  |
| --- | --- | --- |
|  | | |
| COMPUTING PAPER 2 | | **9597** |
| Higher 2  100 marks | | **25 September 2019**  **3 hours** |
|  | |  |
|  |  |  |
|  | | |

|  |  |
| --- | --- |
| **Instructions: Answer all questions.** |  |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| This paper consists of **6** questions. | |
| **[Turn over]** | |

**1.** The following Gantt chart shows the key tasks involved in a data science project for a   
 product recommendation engine based on customers' past purchase patterns.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **AID** | **Activity** | **Week** | | | | | | | | | | | | | | | | | | | | |
| A | Understand the problem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B | Review with team |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C | Make problem statement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D | Define scope of work |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| E | Identify suitable algorithms |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F | Collect data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| G | Clean data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| H | Exploratory data analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I | Develop use cases |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| J | Present use cases |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| K | Analyse full data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L | Develop proof of concept |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| M | Get customer approval |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N | Build final models |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| O | Deploy models |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P | Sign off |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

A project manager often uses both PERT chart and Gantt chart to illustrate and manage a   
 project workflow.

**(a)** Give one benefit and one limitation of using a Gantt chart to depict a project workflow? [2]

|  |
| --- |
| * benefit - quick and easy indication of whether a project is on or off schedule * limitation - does not show as clearly dependent stages as PERT chart |

**(b) (i)** Construct a PERT chart to depict the project work flow. [4]

|  |
| --- |
| * accept sensible possible answers * correctly drawn activities with start and end * correctly labelled time with units in diagram or legend (preferred) * completeness |

**(ii)** State the critical path and the minimum project completion time. [2]

|  |
| --- |
| * A → B → E → H → I → J → K → L → M → N → O → P * 21 weeks |

**(iii)** Explain and give an example of a dependent activity. [2]

|  |
| --- |
| * define dependent activity * example with activity id and activity name |

**(iv)** Explain and give an example of a concurrent activity. [2]

|  |
| --- |
| * define concurrent activity * example with activity id and activity name |

**(v)** Indicate in your PERT chart and justify a suitable dummy activity. [2]

|  |
| --- |
| * indicate dotted line in PERT chart * define dummy activity and give example to show understanding (eg approval) |

**(vi)** Give an example to show your understanding of float or slack time. [2]

|  |
| --- |
| * define float / slack time * example to show buffer or vary start/end times without impacting project completion time |

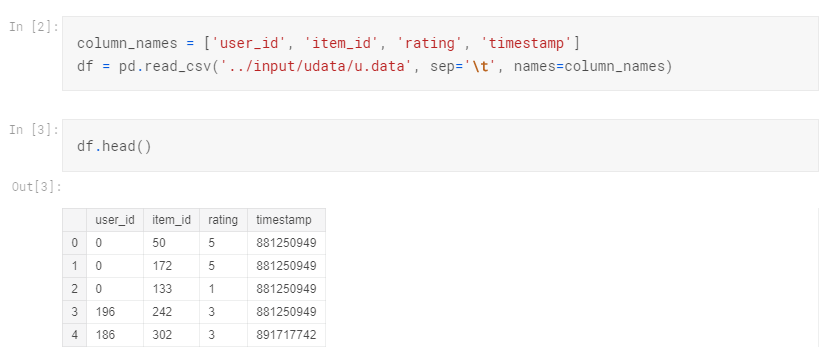
**(c)** The project manager needs to include a documentation activity and a cybersecurity   
 activity to the project. Justify the significance of these activities and show how these   
 can be included in your PERT chart. Explain any implications to the critical path and   
 projection completion time. [5]

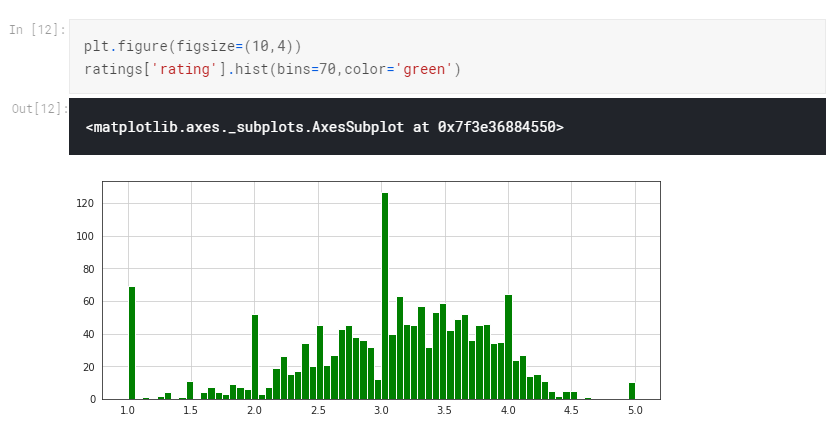
|  |
| --- |
| * documentation activity to keep a record of processes and decisions made for common understanding across project team * cybersecurity activity to ensure system is protected against online malicious attempts / prevent fraud / safeguard data, reasonable duration * documentation can start as early as possible and is generally a concurrent activity * cybersecurity can be near end of project as a dependent activity, reasonable duration * documentation will likely not affect critical path and projection completion time due to its concurrent nature while cybersecurity as a dependent activity can add to critical path and extend projection completion time |

**(d)** The project team would inadvertently have access to some restricted customer purchase  
 information. Give two ethical considerations related to the privacy of data and suggest  
 possible mitigation measures. [4]

|  |
| --- |
| * possible unethical access may lead to sale of customers' preferences and contacts to products/services companies * customers may be bombarded/overwhelmed with unsolicited recommendations/advertising * non-disclosure clause and adherence to data protection act * assure customers of intended usage and data retention period * anonymise by removing identification information (eg customer id, name, address) and aggregate data for reporting as a pool |

The following shows sample interaction snippets of an interactive exploratory data   
 analysis session.





**(e) (i)** State the interface used and justify why this is the most appropriate form of user   
 interaction. [3]

|  |
| --- |
| * interactive scripting interface with tabular data display and charting/plotting facilities * short snippets of program code with immediate output in an organised / visual way * good for investigative and exploratory work to quickly analyse data and relationships / trends |

**(ii)** The data analysis team is deciding on whether to perform the analysis online in a   
 cloud infrastructure or to process all data on a local computer. What are two factors to   
 consider in arriving at this decision? [2]

|  |
| --- |
| * amount of data available for training / processing * amount of storage and processing power available * security of data * timeliness of data |

**(iii)** Evaluate the pros and cons of each approach and make a recommendation with   
 reason(s) to the project team. [5]

|  |
| --- |
| * elaboration on above factors [2] * recommendation with reason(s) [3] |

**(f)** Given the relationship: bit rate = baud rate \* voltage (# bits per signal)

**(i)** Explain the difference between baud rate and bit rate. [2]

|  |
| --- |
| * bit rate is the number of bits transmitted per second whereas baud rate is the number of signal units transmitted per second and one signal unit is able to represent one or more bits * baud rate is therefore always less than or equal to bit rate |

**(ii)** The following voltage levels expressed in volts are chosen to encode bits:

-6.0, -4.5, -3.0, -1.5, +1.5, +3.0, +4.5, +6.0

How many bits represent these voltages? [1]

|  |
| --- |
| * 8 |

**(iii)** For the above voltages, write down one possible set of corresponding bit patterns. [1]

|  |
| --- |
| * 00001111 |

**(iv)** If the baud rate of the line is 900 baud what is the bit rate for the voltage levels? [1]

|  |
| --- |
| * 900 \* x |

**2 (a)** Given an array of integers, devise an algorithm for a function FindLargest that   
 arranges them in order to return the largest possible integer. For example, given [10, 7,   
 76, 415], your algorithm should return 77641510. [5]

|  |
| --- |
| result = ''  convert all numbers to strings  loop while not empty array  find maximum string maxstr  result += maxstr  remove maxstr from array  convert result to integer |

**(b)** Given an array of integers, devise an algorithm for a function MultiplyNotMe that   
 returns a new array such that each element at index i of the new array is the product of   
 all the numbers in the original array except the one at i. For example, given [1, 2, 3, 4,   
 5], your algorithm should return [120, 60, 40, 30, 23]. [5]

|  |
| --- |
| MultiplyNotMe(A)  product = 1  for i from 0 to length(A)-1  product = product \* A[i]  for i from 0 to length(A)-1  B[i] = product  for i from 0 to length(A)-1  B[i] = B[i] / A[i]  return B |

**3 (a)** Given a binary search tree of positive integers, devise an algorithm   
 FindFloorCeiling to find the floor and ceiling of a given positive integer k. The floor   
 Is the highest element in the tree less than or equal to an integer, while the ceiling is   
 the lowest element in the tree greater than or equal to an integer. If either value does   
 not exist, return -1. [5]

|  |
| --- |
| FindFloorCeiling(root, x, floor=Nil, ceil=Nil)  if not root  return floor, ceil  if x = root.data  return x, x  else if x < root.data  floor, ceil = FindFloorCeiling(root.left, x, floor, root.data)  else if x > root.data  floor, ceil = FindFloorCeiling(root.right, x, root.data, ceil)  return floor, ceil |

**(b)** Given a sorted array, devise an algorithm MakeBalancedBST to convert it into a   
 height-balanced binary search tree. [5]

|  |
| --- |
| MakeBalancedBST(array)  if not array  return Nil    mid = len(array) div 2  root = Node(array[mid])  root.left = MakeBalancedBST(array[0:mid-1])  root.right = MakeBalancedBST(array[mid+1:len(array)])  return root |

**4.** Blackjack is a two-player card game whose rules are as follows:

* The player and then the dealer are each given two cards.
* The player can then "hit", or ask for arbitrarily many additional cards, so long as their total does not exceed 21.
* The dealer must then hit if their total is 16 or lower, otherwise pass.
* Finally, the two compare totals, and the one with the greatest sum not exceeding 21 is the winner.

For this problem, cards values are counted as follows: each card between 2 and 10   
 counts as their face value, face cards count as 10, and aces count as 1.

**(a)** Express the above blackjack game flow using a program flowchart. [4]

|  |
| --- |
| * correct program flowchart symbols (include start and end) * correct input/output * correct decision boxes with Y/N paths * correct sequence logic [up to 2] |

**(b)** Given perfect knowledge of the sequence of cards in the deck, devise an efficient   
 algorithm that maximises the player's score (i.e. wins minus losses). [8]

|  |
| --- |
| possible solutions:   * <https://leetcode.com/discuss/general-discussion/292780/MIT-Blackjack-Dynamic-Programming> * <https://github.com/vineetjohn/daily-coding-problem/blob/master/solutions/problem_238.py> |

**(c)** Evaluate the efficiency of your algorithm for part (b). [3]

|  |
| --- |
| * O(n) * explanation [2] |

**5.** A role playing game (RPG) uses object-oriented programming (OOP) to store its game  
 characters' data. A character, either a hero or a monster, has a name, health, magic   
 points and inventory. Each character also has a take\_damage() and a display() method.

The game also has a special type of character, a dragon, which also has additional data   
 airSpeed and breathType.

**(a)** Draw a class diagram showing the relationship between the different game characters.[4]

|  |
| --- |
| * superclass Character and subclass Dragon * appropriate private data and public methods for encapsulation * upward pointing arrow for inheritance * display/show method for polymorphism |

**(b)** Using appropriate examples, explain the following terms:

**(i)** encapsulation

|  |
| --- |
| * definition * suitable example from one class |

**(ii)** inheritance

|  |
| --- |
| * definition * suitable example from subclass in relation to superclass |

**(iii)** polymorphism

|  |
| --- |
| * definition * suitable example from all classes (display/show or any appropriate polymorphic method) |

[6]

**(c)** Using suitable examples, explain why OOP is a preferred programming paradigm in   
 game development than a(n) imperative/procedural one. [2]

|  |
| --- |
| * more realistic model of entities as embodiment of state and behaviour * autonomous message passing mechanism for communication * better protection of data via encapsulation, code reuse via inheritance, code generality via polymorphism |

**6.** BuildingBloCS is Singapore's first/only/largest by Computing students for Computing   
 students and beyond national Computing education outreach programme. In 2019, it   
 comprises a series of workshops, talks, games, projects showcase, programming quizzes,   
 lucky draws and more (media and entertainment very important).

The organisers would like to apply what they learned in Computing to manage workshop   
 information using a relational database.

* Each participant can register for one or more workshops
* Each workshop is conducted by one or more instructors
* Each workshop is also facilitated by one or more facilitators

Due to Personal Data Protection Act (PDPA), it is decided to use an alternative unique   
 identifier as the primary key instead of collecting participants' NRICs/FINs.

The normalised design requires a number of tables.

**(a)** Draw an Entitiy-Relationship (E-R) diagram that shows these tables and the   
 relationships between them. [4]

|  |
| --- |
| * correct entities - Participant, Workshop, Instructor/Facilitator, Training * correct relationships and cardinalities [2] |

**(b)** Suggest and justify a suitable primary key candidate other than NRIC/FIN. [2]

|  |
| --- |
| * email address (assume each participant has its own email address) as this is less individual identifying and not used in national classified/restricted government/citizen services * accept system generated primary key (eg autoincrement integer field) |

**(c)** A table description can be expressed as:

TableName(Attribute1, Attribute2, Attribute3, …)

The primary key is indicated by underlining one or more attributes.

Derive the table descriptions for the tables. [6]

|  |
| --- |
| * UNF - exclude derived/computed fields * 1NF - remove repeating fields/groups * 2NF - remove partial key dependencies * 3NF - remove non-key dependencies * appropriate attribute names * underline primary/composite keys and denote foreign keys by \* or # |

**(d)** There are some fields with missing or null values. Explain how these arise and how a   
 Database Management System (DBMS) may provide facilities to ensure the information   
 is appropriately managed. [3]

|  |
| --- |
| * Some workshops may not need extra hardware/materials/accessories and hence this field can be null * Some participants may not have provided their handphone numbers and hence this field can be updated when available at a later date * DBMS typically will be able to enforce missing or not null constraints during record entry time eg if handphone number is set as not null at schema creation time then the system will prompt with error message if this data is not provided |

**\*\*\* END OF PAPER \*\*\***

**& THANK YOU FOR ORGANISING AN AMAZING BUILDINGBLOCS 2019!**