Name: ……………………………………………….. ( ) Class: ……… Date: ………………….

|  |  |  |
| --- | --- | --- |
| **3.2** | **Algorithm Design** | **Pseudo-code** |

**IF…THEN…ELSEIF…ELSE…ENDIF**

1. Study the algorithm below for calculating weekly pay based on the number of hours worked:

|  |
| --- |
| INPUT hours\_worked  hourly\_rate = 10  IF hours\_worked <= 40 THEN  weekly\_pay = hours\_worked \* hourly\_rate  ELSE  weekly\_pay = (hours\_worked \* hourly\_rate) \* 1.5  ENDIF  OUTPUT weekly\_pay |

a) Write down the output for each of the following input:

1. **hours\_worked** = 40 ……………………………………………………………………………………
2. **hours\_worked** = 41 …..………………………………………………………………………………

b) Amend the above pseudo-code so that the weekly pay is calculated as follows:

|  |  |
| --- | --- |
| Number of hours worked is 40 or less | Use normal hourly rate of $10/hour |
| Number of hours worked is more than 40 | Use normal hourly rate for first 40 hours; then use twice the normal hourly rate for each additional hour of work |

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

2. Write an algorithm, using pseudo-code, that verifies whether an applicant at a driving school is eligible to enrol for driving lessons. The algorithm should prompt the applicant to enter his/her age, then verify whether the applicant is at least 18 years old. Output “You are not eligible to drive” if the applicant is below 18 years old. Output “You are eligible to drive” if the applicant is 18 years old and above.

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

3. A student’s grade is obtained from his/her percentage marks using the table below:

|  |  |
| --- | --- |
| **Marks (percentage)** | **Grade** |
| 75% and above | A1 |
| 70% to 74% | A2 |
| 65% to 69% | B3 |
| 60% to 64% | B4 |
| 50% to 59% | C5 |
| Below 50% | U |

Write an algorithm below, using pseudo-code, which will

* accept a grade as input, then
* output the corresponding range of marks

For example, if the input is “A2”, the output will be “70% to 74%”.

You may assume that the input will always be valid.

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

**WHILE…ENDWHILE**

4. Study the algorithm below:

|  |
| --- |
| Count = 0  Sum = 0  WHILE Count < 5  INPUT Number  IF Number MOD 3 == 0 THEN  Sum = Sum + Number  ENDIF  Count = Count + 1  ENDWHILE  OUTPUT Sum |

1. Complete the following trace table for the algorithm.

Use test data 6, 7, 8, 9, 10 as input.

|  |  |  |  |
| --- | --- | --- | --- |
| **Count** | **Sum** | **Number** | **OUTPUT** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

b) State the purpose of this algorithm.

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

5. Write an algorithm, using pseudo-code, which calculates the total cost that a customer has to pay for shopping at a gift shop. A 15% discount is given on the total cost if the customer is a member.

The algorithm takes the following inputs:

* Whether the customer is a member (‘Y’ for Yes, ‘N’ for No)
* Number of items purchased
* Cost of each item

The algorithm produces the following three outputs:

* Total cost
* Discount
  + Discount = 0.15 × Total cost (for members)
  + Discount = $0 (for non-members)
* Final cost (Final cost = Total cost − Discount)

You may assume that the input is always valid.

Sample 1

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| membership: Y  number: 3  item\_cost: 2.50  item\_cost: 3.00  item\_cost: 3.50 | 9.0, 1.35, 7.65 |

Sample 2

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| membership: N  number: 4  item\_cost: 5.00  item\_cost: 10.00  item\_cost: 15.00  item\_cost: 20.00 | 50.0, 0, 50.00 |

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………  
  
……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………

**FOR…NEXT**

6. The algorithm below sums up the marks of five students for a test.

|  |
| --- |
| total = 0  FOR count = 1 to 5  INPUT mark  total = total + mark  NEXT count  OUTPUT total |

Complete the following trace table for the algorithm.

Use test data 10, 20, 30, 40, 50 as input.

|  |  |  |  |
| --- | --- | --- | --- |
| **count** | **total** | **mark** | **OUTPUT** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

7. Amend the algorithm in question 7 to also output the students' average, highest and lowest mark. You may assume that 50 is the maximum mark for the test.

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………

8. Write an algorithm, using FOR…NEXT, that accepts a positive integer N and outputs the multiples of 5 between 1 and N inclusive.

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

9. Write an algorithm, using FOR…NEXT, that accepts a positive integer N and outputs the result of calculating 12 + 22 + 32 + … + N2.

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

**Reflection questions**

* Give a situation where FOR…NEXT cannot be used.

……………………………………………………………………………………………………………………………………………….  
  
……………………………………………………………………………………………………………………………………………….