

- 4 (a)** A restaurant offers a fixed priced menu. A meal for an adult costs \$9.99 and a meal for a child costs \$6.99.

If there are 6 or more people at a table, a 15% discount is applied to the total cost of the order.

Write an algorithm in pseudocode to:

- input the number of adults and the number of children at a table
- calculate the total cost of the order with the discount, if this applies
- output the final cost of the order.

[5]

- (b)** The final cost of the order after the discount may have more than 2 decimal places.

Write a pseudocode statement to set the final cost of the order to 2 decimal places. You do **not** need to rewrite the whole algorithm.

[2]

(c) The restaurant offers a discount on a meal if the person ordering is either a student or is 65 years or older. The discount (**X** = 1) can only be given if the amount of the order is \$20.00 or over.

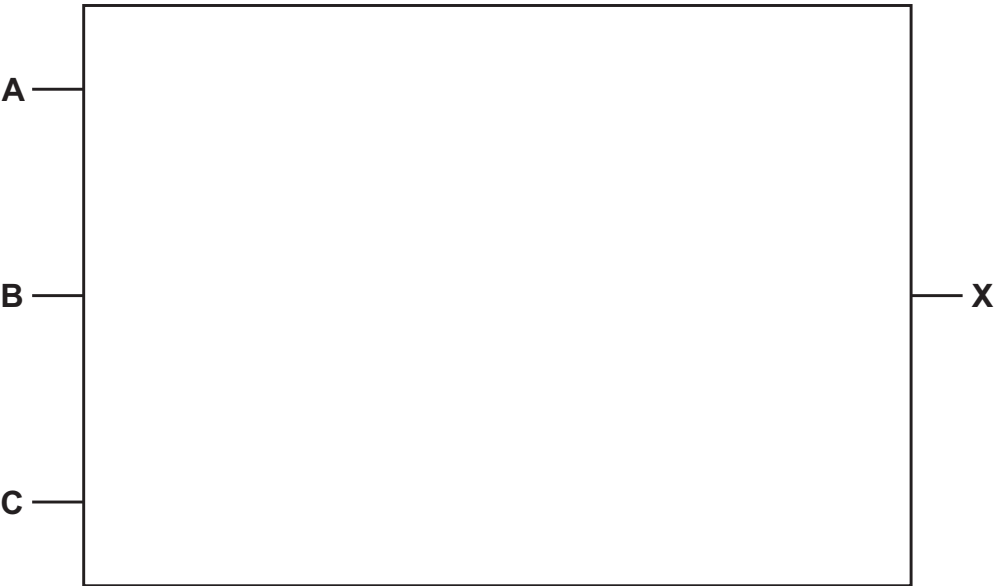
rule A	a student	0
	not a student	1
rule B	under 65 years old	0
	65 years or older	1
rule C	under \$20.00	0
	\$20.00 or over	1

(i) Write the logic expression for this problem.

..... [2]

(ii) Draw a logic circuit for this logic expression.

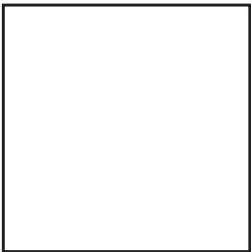
Each logic gate must have a maximum of **two** inputs.
Do **not** simplify this logic expression.



[3]

(d) Draw the NAND logic gate symbol **and** complete the truth table for it.

NAND logic gate symbol:



Truth table:

A	B	X
0	0	
0	1	
1	0	
1	1	

8 A program is required to test the fairness of the random number generator.

The one-dimensional (1D) array `RandomNumber[]` is used to store 100 000 random integers.

The two-dimensional (2D) array `CountedNumber[]` is used to store the integers 1 to 10, inclusive, and the frequency of each integer (the number of times each integer was generated).

Write a program that meets the following requirements:

- Generate 100 000 random integers between 1 and 10, inclusive.
- Store each integer in the appropriate array.
- Calculate the frequency of each of the integers 1 to 10 and store these values in the appropriate array.
- Sort the contents of the counted number array in **descending order** of frequency.
- Calculate the chance of generating each of the integers 1 to 10, rounded to four decimal places.
Use the formula: $\text{chance} = \text{frequency} / 100000$
- Output each of the integers 1 to 10 in the order they are stored in the counted number array, along with the chance of generating them.

The following is an example of an algorithm that sorts a 2D array in an ascending order:

```

F ← TRUE
WHILE F DO
    F ← FALSE
    FOR I ← 1 TO 9
        IF A[I, 2] > A[I + 1, 2]
            THEN
                T1 ← A[I, 1]
                T2 ← A[I, 2]
                A[I, 1] ← A[I + 1, 1]
                A[I, 2] ← A[I + 1, 2]
                A[I + 1, 1] ← T1
                A[I + 1, 2] ← T2
            F ← TRUE
        ENDIF
    NEXT I
ENDWHILE

```

You must use pseudocode or program code **and** add comments to explain how your code works.

All arrays, variables or constants used must be declared for this algorithm.

All inputs and outputs must contain suitable messages.

3

[illegible]

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

- 8 A programmer is designing a program to check the length of a password and to check if the password input is the same as the stored password.

The program requirements are:

- input the password, `Password`
- check if there are at least 8 characters in the password
- check that the password is **not** the same as the stored password `OldPass`
- output 'accepted' if both tests are completed successfully
- otherwise, output 'rejected'.

Use the variable names given.

- (a) Complete the flowchart for the program.



(b) The accepted password, `Password`, is to be written to the file `MyPassword.txt`

Write pseudocode to:

- open the file
- write the accepted password to the file
- close the file.

[3]

(c) Explain why the accepted password needs to be stored in a file.

[2]