

**93604**



**Mana Tohu Mātauranga o Aotearoa**  
New Zealand Qualifications Authority

## **Scholarship 2025**

### **Digital Technologies**

**This assessment has THREE questions. Attempt ALL questions.**

## INSTRUCTIONS

This assessment has THREE questions, each relating to a different programming problem. You should attempt all three questions.

Each question has several parts. These must all be answered in as much detail as possible.

Code can be written in:

- Pseudocode
- Python 3
- C++
- C
- C#
- Java
- JavaScript

Code cannot be run, so care must be taken to manually debug and check accuracy. The markers will take appropriate steps to ensure this is considered.

Before you begin, ensure that you have Resource Booklet 93604R.

## QUESTION ONE

Consider the following pseudocode:

```
data = [8,5,7,4,1,2,6,3,9]
```

```
n = length(data)
```

```
FOR i FROM 0 TO n - 1:
```

```
    FOR j FROM 0 TO n - 2:
```

```
        IF data[j] > data[j + 1] THEN:
```

```
            swap data[j] with data[j + 1]
```

```
        ENDIF
```

```
    ENDFOR
```

```
ENDFOR
```

```
PRINT data
```

(a) What is the final output of this algorithm?

(b) What is the purpose of this algorithm?

(c) How many comparisons would it take on a list of 100 items in random order?

(d) How many comparisons would it take on a list of 1,000 items in sorted order?

(e) What is the cost of this algorithm, and how did you come to this conclusion?

- (f) What would the implications be for the cost of this algorithm when run on a list of 1,000 items that all had the same value?

- (g) What are the implications, both positive and negative, of using this algorithm in a real-world setting?

- (h) Write a more efficient algorithm to achieve the same purpose.

data = [8,5,7,4,1,2,6,3,9]

- (i) Evaluate the efficiency of your algorithm compared to the original one.

## QUESTION TWO

Consider the following problem.

You are given a grid of characters of dimensions N by N. Calculate the fastest way to get from the top left corner to the bottom right corner.

You may only travel through grid positions that contain a '\*' symbol and must avoid positions that have a '#' symbol. In each step you can travel right, left, up or down, but not diagonally.

You can assume that there will always be at least one available path to the destination.

Sample input:

```
grid = [  
    ['*', '*', '#', '*'],  
    ['*', '#', '*', '*'],  
    ['*', '*', '*', '#'],  
    ['#', '*', '*', '*']  
]
```

Output:

6

(a) Provide a second, larger, input and expected output data set for this problem.



- (b) Describe the route your second sample takes that gives the provided output.

- (c) Explain how you might go about solving this problem.

- (d) Write a solution to the above problem in your chosen language.

(e) What data structure(s) did you use to solve the problem? Justify your decision(s).

(f) Explain, with examples, how your approach would scale to larger grid sizes.

(g) What would **your** solution return if the sample data did not have a viable path to the target? Discuss.

- (h) What if the # symbols could be traversed, but they would 'cost' a value of 4. How would this impact your approach?

## QUESTION THREE

Consider the following problem.

### Perfect pastry packing

You run a bakery that sells pastries that are all the same size. These pastries are supplied to customers in different sized boxes. Calculate the smallest number of boxes that you can use to perfectly pack a number of pastries.

You are given an array of boxes where `boxes[i]` represents how many pastries can be stored in that size of box. You have an unlimited supply of boxes. You will be given a number `N` that represents how many pastries you need to pack.

Write a function that returns the minimum number of boxes needed to perfectly pack `N` pastries. If it is not possible to perfectly pack `N` pastries, the function should return `-1`.

#### Example 1:

`boxes = [3, 5, 7]`

`N = 11`

Output:

3

Explanation: The best way to pack 11 baked goods is using  $5 + 3 + 3$  (3 boxes in total).

#### Example 2:

`boxes = [2, 4]`

`N = 7`

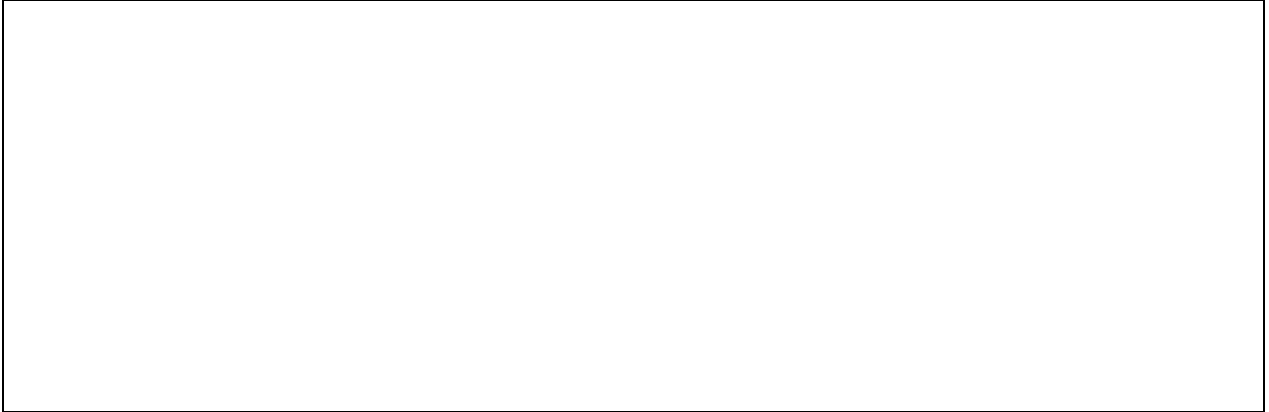
Output:

-1

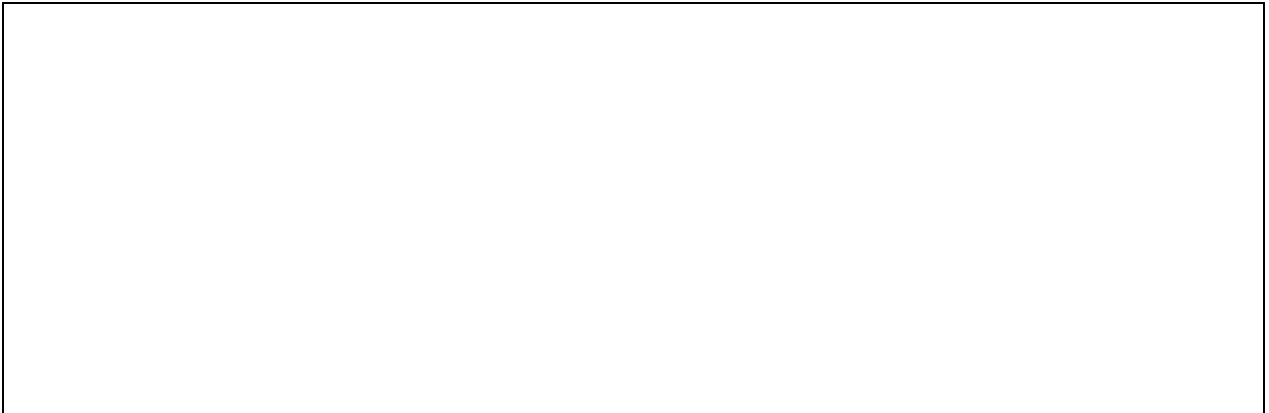
Explanation: It is not possible to pack exactly 7 baked goods with boxes of size 2 and 4.

(a) Describe your approach to solving this problem to ensure you have the smallest number of boxes.

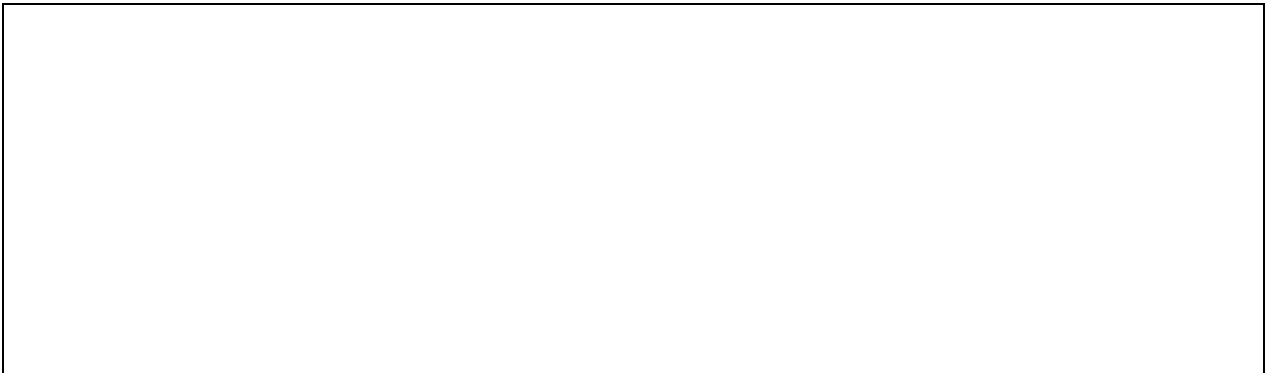
(b) Write a solution to this problem in your chosen language.



(c) Explain how your solution solves the problem.



(d) Analyse your solution in terms of its efficiency.



- (e) Given the examples supplied, explain why a 'greedy' solution would not provide an accurate result.

- (f) This problem could be solved in several ways. Describe how else it could be solved, and evaluate your solution against those other methods.