SUPERVISOR'S USE ONLY

S

93604

SAMPLE

Draw a cross through the box (☒) if you have NOT written in this booklet

+

NZQA | Mana Tohu Mātauranga o Aotearoa
New Zealand Qualifications Authority

Scholarship 2025
Digital Technologies

Time allowed: Three hours
Total score: 24

SAMPLE ASSESSMENT

This sample assessment contains three questions that are intended to give some guidance regarding the potential format, expectations, and problem-solving approaches that candidates could employ in the final assessment.

The final assessment will only be accessible through an NZQA-provided online portal. It will contain three questions, and candidates will be expected to attempt all three. Text entry boxes will expand as necessary to fully accommodate candidates' anwers.

For the final assessment, blank paper will be provided for rough working, but not for submission as part of the examination. A resource sheet will also be provided.

Question	Score
ONE	
TWO	
THREE	
TOTAL	

ASSESSOR'S USE ONLY

INSTRUCTIONS

You should attempt all three questions.

Code can be written in:

- Pseudocode
- Python
- 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.

Each programming problem includes several long-answer questions. These must be answered in as much detail as possible.

QUESTION ONE

Consider the following pseudocode:

```
FUNCTION DoSomething(a):
    b = “”
    FOR i FROM (LENGTH(a) - 1) TO 0 STEP = -1
        b = b + a[i]
    ENDFOR
    answer = true
    FOR j FROM 0 to LENGTH(a):
        IF a[j] IS NOT EQUAL TO b[j]
            answer = false
        ENDIF
    ENDFOR
    return answer

PRINT DoSomething(“Scholarship”)
```

(a) What is the output of this pseudocode?

(b) What is the purpose of the “DoSomething” function, assuming that ‘a’ is a string?

(c) Write three test cases for this function that return true.

(d) Write three test cases for this function that return false.

(e) Write one boundary case for this function.

(f) Write the improved pseudocode function to handle this boundary case in the box below.

(g) Evaluate the efficiency of this algorithm and suggest improvements.

(h) Assuming that improvements can be made on the efficiency of this algorithm, in the box below, write the most efficient algorithm you can think of to achieve the intended purpose of the "DoSomething" function.

Note: You are not allowed to use reverse() *or sort*().

(i) Explain your improved algorithm and evaluate its efficiency against the original supplied algorithm.

QUESTION TWO

Consider the following problem.

You are given a challenge to calculate the minimum number of moves a knight takes to get from a target position to a destination. You are given the size of the chess board and a series of queries. Each query lists two squares on the board, the starting position and the target position. You are required to give the minimum number of moves needed for a knight to move between these positions. As a reminder, the knight moves in an L shape, moving one square in one direction and two in the other; the knight cannot move out of the board.

Input

The first line of input will be single integer **N**

denoting a chess board of size **NxN** where **4≤N≤100**.

The following lines will contain a series of queries, of the form

a,b,c,d where **(a,b)** and **(c,d)** are the two squares you need to move the knight between.
Note that **1≤a,b,c,d≤N**

The input will be terminated by the line:
0 0 0 0

There will be at most 100 queries per test case.

Output

For each query you should output a single integer, the minimum number of knight moves to traverse between the two squares.

Sample input

8
1 1 1 2
4 4 5 6
1 1 8 8
0 0 0 0

Sample output

3
1
6

(a) Explain how it would take the knight three moves to get from position 1,1 to position 1,2 in the first part of the sample data.

(b) Assuming that all start and end positions are on the board, is there ever a situation where the knight cannot reach the target?

(c) Create another sample to test a solution against that includes boundary input.

(d) Write a solution to this problem in your chosen language in the box below.

(e) Discuss the approach you chose to solve the problem of finding the minimum number of moves for a knight to travel between two squares on a chess board.

(f) State and explain the cost of your algorithm on inputs of increasing size.

(g) Compare your chosen method with a potential alternative solution, and describe how the new approach would compare regarding efficiency and complexity.

Please turn over ➤

QUESTION THREE

Consider the following problem.

San Holo and his friend Bew Chaka need your help again. They've just dropped out of hyperdrive in the Gollius System and there are several trade stations that they can go to. The problem is, they want to maximise the amount of profit while keeping their cargo weight less than or equal to the maximum. Your task is to quickly work out the maximum amount of profit from any combination of trades at a particular trade station. The trade items **cannot** be split up, and each item can only be included once. You must evaluate the maximum profit that can be made from trading at the given station from the supplied data.

Input

The first line of input is two integers, **N** and **M**. **N** is the number of possible trade routes that will follow, and **M** is the maximum cargo space left on the ship. There then follows **N** lines, each line containing an item's weight **W** and the profit **P** that it will generate when sold back at base.

Constraints

$1 \leq N,M \leq 1000$

Output

Output will be a single line containing one integer value for the maximum profit that you can achieve at this station.

Sample input

3 5
1 3
2 5
5 7

Sample output

8

(a) Explain how this sample produces the output of 8 for the given input.

(b) Create one more sample input and output set in the box below.

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

(d) Write a solution to this problem in your chosen language in the box below.

(e) Discuss any challenges you faced in creating a solution to this problem.

(f) Explain how your solution will scale to larger input sets.

(g) Compare and contrast your chosen solution with other methods that could have been used to solve this problem, and justify your choice of approach.

Extra space if required.
Write the question number(s) if applicable.

QUESTION NUMBER

Extra space if required.
Write the question number(s) if applicable.

QUESTION NUMBER

Extra space if required.
Write the question number(s) if applicable.

QUESTION NUMBER