This is the First Round of the Swiss Olympiad in Informatics. The test consists of 16 multiple choice questions, each worth 1 point.

Rules: You must complete this exam by yourself and without AI. Calculators are allowed, but not necessary.

General Questions (10 questions): Contains questions involving logical reasoning and simple informatics ideas.

Program Questions (6 questions): Contains questions with easily understandable psuedocode and some questions about it. Prior programming knowledge is not required.

Important: The problems in each section are sorted by difficulty. Remember to try problems from both sections.

Time Limit: 40 minutes

Duration: 40 minutes

1 General Questions

These are the general questions. The following 10 questions are ordered by difficulty. All questions are worth 1 point. There is no penalty for wrong answers.

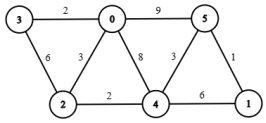
Question 2

Given a sequence $[5, 2, 1, 3]$, what is the minimum number of steps required to sort the sequence in ascending order? In each step, you may swap two adjacent elements. For example, $[5, 2, 1, 3]$ can become $[5, 2, 3, 1]$ in one step, but becoming $[1, 2, 5, 3]$ in one step is not possible.

A. 3 B. 4 C. 5 D. 6 E. 7

Question 3

There are 6 cities (numbered $0, 1, 2, 3, 4, 5$) connected by two-way roads. Each road connects two cities and takes a certain amount of minutes to travel, as indicated in the diagram above. What is the minimum amount of time it takes to travel from city 3 to city 1?



A. 10 B. 11 C. 12 D. 13 E. 14

Question 4

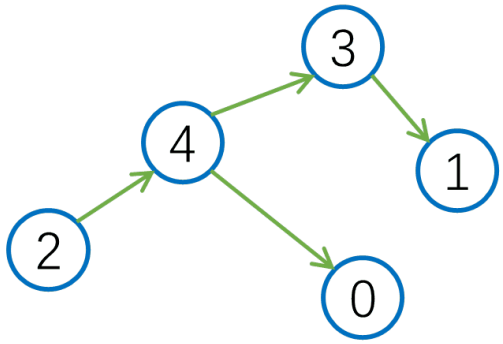
You have 9 yellow ducks and 99 white ducks. You may buy an extra yellow duck for 1 CHF and a white duck for 1 CHF. You may paint any duck red for free. What is the minimum number of CHF that you need to pay to have the same number of ducks of every color, given that you cannot get rid of ducks?

A. 39 CHF B. 42 CHF C. 69 CHF D. 90 CHF E. 189 CHF

Question 5

There are 5 cities (numbered $0, 1, 2, 3, 4$) connected by one-way roads. The map above shows the current roads between cities. It is not possible to start from one city and visit every other city exactly once using the roads shown.

Which one of the following new roads, if added to the map, would make it possible to visit all the cities exactly once, starting from one city and following the direction of the roads?



A. $0 \to 1$ B. $0 \to 3$ C. $2 \to 0$ D. $2 \to 3$
E. $4 \to 1$

Question 6

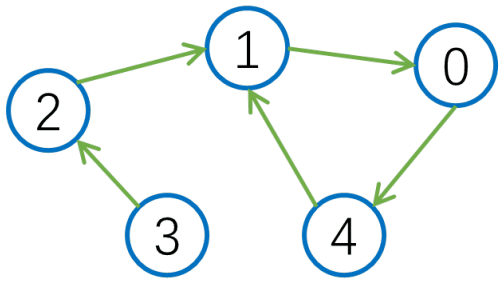
Mouse Stofl has selected a number between 1 and 25 and is not telling it to you. You may guess a continuous sequence of numbers (for example $4 - 7$), and Mouse Stofl will tell you if his number is in your range, above, or below. What is the minimum number of guesses you need to be sure of Stofl's number, in the worst case?

A. 3 B. 4 C. 5 D. 6 E. 7

Question 7

There are 5 cities (numbered $0, 1, 2, 3, 4$) connected by one-way roads. You are allowed to add one new one-way road between two cities where there isn't already a road between them (in particular, it is not possible to add a road between two cities connected by a road in the other direction) .

How many different ways are there to add such a road so that, after adding it, it becomes possible to travel from any city to any other city by following the direction of the roads?



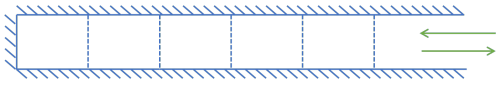
A. 0 B. 1 C. 2 D. 3 E. 4

Question 8

There is a narrow parking lot with 5 parking spaces in a straight line. Notably, a car can only leave if there's no other car parked between it and the exit. Five cars — A, B, C, D, and E — enter the parking lot in that order (A first, then B, and so on).

Note: Cars may leave before all five cars enter the parking lot.

Which of the following exit orders is not possible?



A. A, B, C, D, E B. B, A, D, C, E
C. C, B, E, D, A D. E, D, C, B, A
E. B, C, E, A, D

Question 9

You are given a list of 4 distinct numbers. You can only compare two numbers at a time to determine which one is larger. Regardless of the initial order of the numbers, what is the minimum number of comparisons required to guarantee that the numbers can be fully sorted in ascending order?

A. 3 B. 4 C. 5 D. 6 E. 7

Question 10

The country of Crazystofl is made up of 2026 islands numbered from 0 to 2025. It is known that from every island x there exists a flight going both ways from $\lfloor \frac{x}{2} \rfloor$. What is the minimum number of flights that one needs to take to go between island number 125 and 1025?

Comment: $\lfloor y \rfloor$ is the largest whole integer which does not exceed the value of y. For example $\lfloor 3.14 \rfloor = 3$, $\lfloor 7 \rfloor = 7$, $\lfloor \frac{11}{2} \rfloor = 5$.

A. 10 B. 11 C. 12 D. 14 E. 16

Question 11

How many ways are there to entirely cover a 4×14 grid with 1×4 tiles such that no tiles overlap? Note that the tiles may be rotated.

A. 43 B. 44 C. 50 D. 55 E. 59

2 Program Questions

These are the program questions. The following 6 questions are ordered by difficulty. All questions are worth 1 point. There is no penalty for wrong answers.

Question 12

In the following program, what is the output of the program?

Notes

- **Set** assigns a value to a specific variable name.

- **If**, **Else If**, or **Else** are called conditionals. Conditionals perform a certain block of code if the condition (specified on the same line) is true. Indentation shows which lines of code belong to each conditional. If none of the conditions are met, the block of code under **Else** will be performed.

- **Print** outputs the value of a variable.

```
Set originalScore ← 76
Set bonus ← 0

If originalScore ≥ 90
    Set bonus ← 5
Else If originalScore ≥ 75
    Set bonus ← 2
Else
    Set bonus ← 0

Set finalScore ← originalScore + bonus
Print finalScore
```

A. 74 B. 76 C. 78 D. 81 E. 83

Question 13

In the following program, what is the output of the program?

Notes

-A **For** loop repeats its code a set number of times, as indicated by the range inclusively. For example, **For x from 2 to 4** would iterate the code three times. First for x = 2, then x = 3, and finally x = 4.

```
Set sum ← 0
For i from 1 to 3
    Set sum ← sum + i * i
Print sum
```

A. 3 B. 6 C. 9 D. 14 E. 27

Question 14

In the following program, what is the output of the program?

Notes

- A **While** loop will continue running **as long as its condition remains true**. The loop checks its condition **at the beginning** of each iteration. Once the condition becomes false, the loop stops, and the program continues with the next instruction after the loop.

- $x == y$ will be considered as true if x and y are equal and false in any other case. For example, 1 == 2 is false and 5 == 5 is true.

```
Set N ← 2026
Set M ← 1

While (M == 1)
    If N < 0
        Set M ← 0

    Set N ← N - 2


Print N
```

A. −4 B. −3 C. −2 D. −1 E. 0

Question 15

In the following program, what is the output of the program?

Notes

-A **Function** defines a block of code which can be called several times. It usually contains an input (described in the parentheses following the function name) and a return value (described in the **Return** clause).

- **a mod b** means remainder when a is divided by b, e.g. 7 mod 3 = 1 and 8 mod 4 = 0.

```
Function foo(x)
    If x mod 50 == 0
        Return x / 50
    Else
        Return foo(x - 1) * foo(x - 1)

End Function


Print foo(99) + foo(100) + foo(101)
```

A. 0 B. 7 C. $2^{2^{48}} + 6$ D. $2^{49} + 2^{50} + 2^{51}$
E. None of the above

Question 16

In the following program, what is the output of the program?

Notes

- **Indices** are position numbers that point to the position of something in a list. They are 0-Indexed. For example, if list = $(2, 4, 6)$ then list[0] = 2 and list[2] = 6.

```
Function CountSafeSquares(n, m, rooks)
    Set count ← 0
    Set rookCount ← length of rooks

    For i from 0 to n - 1
        For j from 0 to m - 1
            Set safe ← true

            For k from 0 to rookCount - 1
                Set rx ← rooks[k][0]
                Set ry ← rooks[k][1]

                If i == rx or j == ry
                    Set safe ← false
                    Break

            If safe == true
                Set count ← count + 1

    Return count

End Function

Print CountSafeSquare(10, 15, [(2, 3), (5, 7), (5, 3)])
```

A. 84 B. 104 C. 112 D. 126 E. 150

Question 17

How many different values can K take initially (yellow field) such that our program outputs "Yay!" for some freely chosen N (yellow field)?

Comment: N and K must be integers.

```
Set M ← 7
Set N ← ____
Set K ← ____

If(K < 0)
    Set K ← 0

While (K < 6)
    Set M ← M + N
    Set K ← K + 1


If (M == 42) Print "Yay!"
```

A. 0 B. 1 C. 2 D. 3 E. Infinitely many possible values