2022

British Algorithmic Olympiad

EXAM PAPER

INSTRUCTIONS

- Please use extra sheets for working. These must be with the paper when collected by your invigilator, and will be marked. Cross out anything you wish not to be marked.
- You may use a calculator.
- You must not use mobile phones, and you may **only** use computers for writing code.
- Do not communicate with anyone other than your invigilator during the exam.

INFORMATION

- Answer **all** the questions. You may answer them in any order.
- There are four questions in this question paper.
- The marks for each question are shown in brackets.
- Some written questions award working marks; we recommend you write all your working.
- Sample cases given for programming questions are **not** included in the mark scheme.
- The questions are designed to be challenging. Questions one and two are a mixture of mathematics and computer science. Question three is focused on mathematics whilst question four is focused on computer science.
- The time limit in which answers to programming questions must terminate is **1 second**.

Name					
Candidate Number					
Total Score			100		

Time Allowed
2 hours 30 minutes

Question 1. Binary Rotors

A *binary rotor* consists of two binary strings of length L $(2 \le L \le 2^6)$. The first binary string is called the *shift*, and the second the *check*. On a single turn of the rotor, the shift is rotated r times $(-2^{46} \le r \le 2^{46})$, where a positive rotation means the first digit is moved to the end of the shift, and a negative means the last digit is moved to the start of the shift. After the i^{th} turn, the i^{th} value of the check (from the left) is modified depending on the first and last value of the shift (we'll call these s_0 and s_1, respectively). To *use* a rotor means to turn it L times.

A XOR gate takes inputs, all either a 0 or a 1. If the number of 1s in the input is odd, it returns 1. Otherwise, it returns 0. XOR(a, b) means to apply a XOR gate to inputs a and b.

The following function, C, can be used to determine the i^{th} value of the check (c_i) after the i^{th} turn:

$$C(s_0, s_1, c_i) = \mathrm{XOR}(\mathrm{XOR}(s_0, s_1), c_i)$$

For example, if we have a binary rotor: $\begin{bmatrix} \text{shift} = 1010010 \\ \text{check} = 0000001 \end{bmatrix}$, where $r = 1$, after the first turn, the rotor will be: $\begin{bmatrix} 0100101 \\ 1000001 \end{bmatrix}$.

A (14 marks)

Write a program that takes three inputs: (in order) the shift, the check and r. Output the final check string, having turned the rotor L times.

Example Run

```
101
000
1
110
```

B (1 mark)

Which of the following provide(s) the correct functionality for XOR(a, b, ab)?

I	$\mathrm{XOR}(a, b, ab) = a + b - ab$	**III**	$\mathrm{XOR}(a, b, ab) = a - (b - ab - 1)^2$
II	$\mathrm{XOR}(a, b, ab) = ((ab)^2 + a - b)^2$	**IV**	$\mathrm{XOR}(a, b, ab) = a(1 - b)^2 + b$

C (4 marks)

Suppose all possible shift values of length 2^{23} are generated, each string containing exactly two 1s. When each is used with a rotor $(r = 1)$ and a check containing only zeros, how many output a final check string consisting of an even number of 1s?

D (6 marks)

Given that S_i is the denary (base 10) value of the shift of the rotor R_α $(r = 1)$ after the i^{th} turn, $\Gamma(x)$ is the remainder of x upon division by 4, and the starting shift of R_α is 110011, find the lowest value of $m \in \mathbb{Z}^+$ such that:

$$mL \sum_{n=1}^{\infty} \left(\frac{\Gamma(S_n)}{2L} \right)^n \in \mathbb{Z}$$

Question 2. Sonar

A new fleet of British naval ships has been built and will soon be deployed in the Atlantic Ocean. These new ships have been designed specifically to protect their crews at night, when the ship is stationary. Each is fitted with a certain number of depth charges which detonate when they detect a nearby submarine or ship. The ships all have a sonar system which is highly effective, the only problem is that the crews lack the knowledge of how many charges to deploy each night to maximise resources and protection. Each ship has a *protection level*, p, which determines the safety that the ship requires each night. Ships with a higher level will have more charges on board.

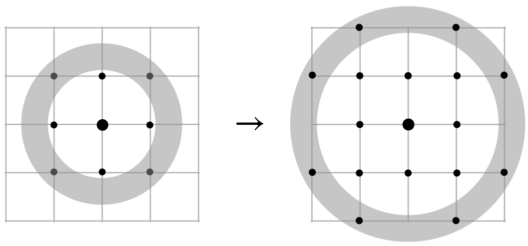
An intelligence team in London has determined the most effective way of deploying charges using the *nautical grid positioning system* (NGPS). This system relies on the idea that the area of ocean that surrounds a ship can be modelled as a grid of square areas of water. Each ship has its own respective $p \times p$ grid. The top left of the grid has coordinates $(0, p)$.

The team concluded that each crew should keep to the following instructions:

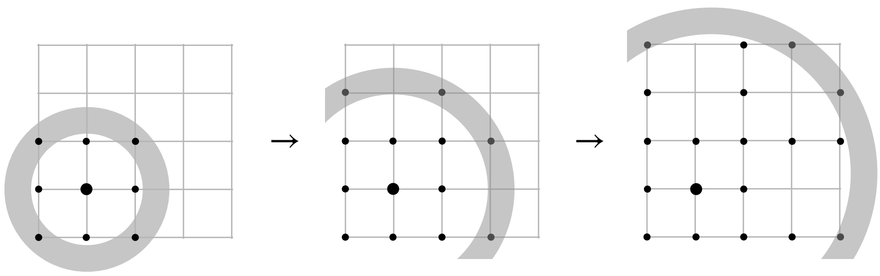
- Use the sonar to emit a signal from the ship, which is located at a certain coordinate within the NGPS.
- Place a depth charge, C, where the signal crosses a vertex of the NGPS grid.
- Do not place any more charges directly behind C.

For example:

When a ship with $p = 4$ and coordinates (2, 2) sends out a signal, it first hits 4 inner vertices, then another 4 vertices. Following, it hits 8 new vertices as shown. The total number of required charges in this example is 16.



The next example shows a similar situation, but where the coordinates of the ship are (1, 1). The total number of charges required here is 18:



A (18 marks)

Write a program for the ships to use that finds the number of charges required for a ship to deploy on a given night. The first line of input will be the value of p $(2 \leq p \leq 2^9)$, and the second will be the coordinates of the ship, $x \in \mathbb{Z}$ followed by $y \in \mathbb{Z}$.

Example Run

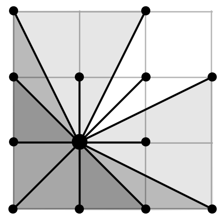
```
4
2 2
16
```

B (2 marks)

A ship S has coordinates (0, 0). The point Q has coordinates (987654321, 123456789). Assuming the value for p is large enough, determine whether a depth charge would be placed at Q. Explain your answer.

C (5 marks)

A new ship H ($p = 999$) is located at (1, 1). Suppose you were to draw straight line segments from the ship to each deployed depth charge; a *nautical triangle* is a triangle formed by the grid boundaries and the line segments. For example, the nautical triangles in this 3×3 grid have been highlighted.



Determine the number of nautical triangles for the ship H.

Question 3. Pairs and Triples

Pythagorean triples are positive integral solutions (a, b, c) to the equation $a^2 + b^2 = c^2$, for example (3, 4, 5) and (15, 36, 39). The first of these is an example of a *primitive* Pythagorean triple as no positive integer divides a, b and c, whereas 3 divides a, b and c in the second example.

Although this theorem, as it relates to right-angled triangles, is attributed to the Greek mathematician Pythagoras (fl. 530BC), it has been found that people were aware of it and these triples as early as 1800BC in Babylon.

A (2 marks)

Show that for integers x, y, the integer $x^2 - y^2$ is odd or a multiple of 4.

B (8 marks)

Suppose (a, b, c) is a primitive Pythagorean triple, *i.e.* $a^2 + b^2 = c^2$ where a, b and c are all integers that have no factor all in common. As *at least* one of a or b is odd, let us assume that b is odd.

Let $\frac{c+b}{a} = \frac{m}{n}$, where $\frac{m}{n}$ is a fraction in its lowest terms. First find $\frac{b}{a}$ in terms of m and n, and using this find a, b and c in terms of m and n. Verify that any positive integer pair (m, n) provides a solution to the equation $a^2 + b^2 = c^2$.

C (11 marks)

Write a program to calculate $c(n)$, the number of pairs of non-negative integers α, β, such that $\alpha^2 + \beta^2 < n$, where $2 \le n \le 2^{36}$. The single input will be the value for n.

Example Run

```
3
4
```

D (1 mark)

State, without proof, the limit of $\frac{c(n)}{n}$ as $n \to \infty$.

E (3 marks)

Suppose $c_1(n)$ is the number of pairs of integers α, β such that one of α and β is negative, the other non-negative and $\alpha^2 + \beta^2 < n$, where n is **not** a perfect square. A 'ceiling' function can be written as $\lceil x \rceil$, which denotes the smallest integer not smaller than x. Given that $t(n) = \lceil n \rceil$, construct an expression for $c_1(n)$ in terms of the functions c and t.

Question 4. Wolves

It's a snowy day on Mount Olympus. A new species of wolf has been occupying the surrounding areas; researchers have named this species the *Lexicon-Wolf* (L-Wolves for short).

L-Wolves have a specific hunting style. They form a straight line next to each other at evenly spaced distances. For each wolf, they work out and organise a unique endpoint on a similar straight line several miles away, so they can most efficiently hunt prey in the area.

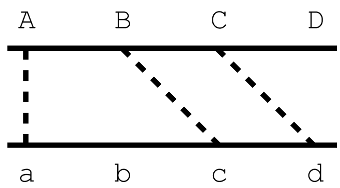
The L-Wolves are very stubborn, however, and will not change their direction having chosen a unique endpoint before the hunt. Although research has shown them to be one of the smartest wolf species, whilst hunting they will kill anything they sense near them, and will follow any tracks they pick up. As such, they cannot cross paths with another wolf during a hunt. To overcome this, the wolves send out groups at different times where no paths overlap. We call one of these groups a *pack order*.

L-Wolves are labelled from left to right alphabetically using uppercase letters and the endpoints similarly using lowercase letters.

For example, if there are four wolves `A`, `B`, `C`, `D` and their four chosen endpoints `a`, `c`, `d`, `b`, respectively, the possible pack orders are:

`A AB ABC AC AD B BC C D`

A diagram of the pack order `ABC` is shown below.



A (18 marks)

Write a program that inputs the endpoints of w wolves ($1 < w < 27$), where the first endpoint is for wolf `A`, the second for wolf `B`, etc. Your program should output the number of possible pack orders followed by the maximum possible number of wolves in a single pack order.

Example Run

```
acdb
9 3
```

B (2 marks)

Suppose a pack of 8 wolves chose the endpoints `abcdefgh` in some order. What is the average of the maximum number of wolves that can hunt over all permutations of these endpoints? Give your answer to 2 decimal places.

C (5 marks)

The wolves in part **B** are trying to remember which permutation of `abcdefgh` they originally agreed upon. To keep their hunting strategy secret, they speak in some sort of code. The following is said:

`A:`	barK.	`E:`	baRK_1.
`B:`	bArK_1 / bArK.	`F:`	bArk_{-2} / baRK / bark_1 / bArK.
`C:`	Bark.	`G:`	bARK / bARk / barK.
`D:`	bark_2 / bARK.	`H:`	baRK_{-1}.

Suggest and explain an original list of endpoints.

END OF OLYMPIAD Total Marks: **100**