Pseudocode for Array Implementation

public static int ChildrenRonde(int[] C, int s)

BEGIN

INIT temp = -1

INIT iter

INIT j

INIT k = 0

WHILE (k < C.length – 1)

iter = 1

WHILE (iter <= s)

temp++

IF(temp >= C.length)

temp -= C.lenght

END IF

IF (C[temp] EQUALS 0)

iter++

END IF

END WHILE

C[temp] = -1

k++

END WHILE

FOR (i = 0; i< C.legnth; i++)

IF (C[i] EQUALS 0)

RETURN i + 1

END IF

END FOR

RETURN -1

END

Pseudocode for Linked List implementation

public void ChildrenRonde(int s)

BEGIN

Node<T> iter = start

IF (size >= 2)

FOR (int i = 1; i< s; i++)

iter = iter.getNext();

END FOR

removeNode(iter)

END IF

WHILE (size > 1)

FOR (int i = 0; i < s; i++)

iter = iter.getNext()

END FOR

removeNode(iter)

END WHILE

PRINT(“There is only one child remaining”)

THIS.print();

END

Test Cases

The two test cases are the examples from the lab document, with size (the first input) being 5 and step size (the second input) beign 2. This will give an answer of 3. The second test case is going to be with a size of 7 and step size of 3, giving an answer of 4. These are two standard run cases so they should work as intended.

The third test case is going to be with a size of 10 and a step size of -1. Since the step size doesn’t make sense in the context of the problem, the program will not execute and it will tell the user that something is wrong.

The final test case is with a size of 10 and a step size of 13. The step size is bigger than the amount of children, however the program will still work since it would still make sense in the real life version of this problem, you can easily just count over the amount of children since they’re in a circle.

Historical Event

While I am not entirely sure what the historical context is that inspired this problem, I would guess that it has to do with the plague in Europe.