Please refer to the online lesson plan on the DT Hub to access all website links and additional resources.

## Learning hook

In order to bring in a level of curiosity, you could start with a very open question like “Is anything truly random?”

1. Explain to students that they will, in real time, create a spreadsheet that simulates the random tossing of a coin. (While doing, this, draw attention to the spreadsheet features students will be using – see the information under 'Teacher background'.)
2. Discuss random numbers and the challenge that a programmer faces in generating them. At this point, as an aside, discuss the vexed issue of randomness: are random functions as used by computers truly random? Is a truly random function possible? Is a physical coin toss truly random? How reliable are the built-in random functions such as those below?
3. Next, students respond to your questions and observe a demonstration, producing an identical spreadsheet as the lesson progresses. The class provides suggestions at appropriate steps.

The process for creating the spreadsheet is detailed in the 'Method' section below. You can 'cue' students to develop solutions at points marked \*.

**Method**

\*Ask students to locate a random function in their spreadsheet application:

**RAND()**

This function generates a decimal number anywhere between 0 and 1

Students will also use:

**=IF(Cell>=0.5, "Heads","Tails")**

**The steps**

1. Type title Coin toss in Cell A1.
2. Add successive column headings, commencing in A2: Toss, Random, Coin, Counts.
3. Type title *HEADS* in D3.
4. Type title *TAILS* in D4.
5. Type title *TOTAL* in D5.
6. Explain that these indicate the cells where the totals of the outcomes will be stored for each toss.
7. Show how to format these cells with background colours.
8. Enter '1' in A3.
9. \*We need to type the integers 1 to 100 down the column. Ask students to suggest a quick method of achieving this (ie enter formula **=A3+1** in Cell A4).
10. Highlight this formula in A4 and *Edit>Fill* down to Row 102 (thus creating 100 coin tosses).
11. \*Ask for suggestions of how we should use the Random formula. Ask: 'What range of output does it produce?' Work with the class to deduce how this could be used to randomly produce one of two random outputs: an H or a T.
12. Enter formula **=RAND()** in Cell B3.
13. Highlight B3 and*Edit>Fill* down to Row 102.
14. \* Using class discussion establish the use of the IF formula as a binary selection based on a test case. Make sure the class understands this formula's structure and logic.
15. After the class establishes the following, enter this formula in C3: **=IF(B2>=0.5, "Heads","Tails")** This creates the outcome of Heads or Tails based on the random number generated in B3.
16. Highlight C3 and *Edit>Fill* down to Row 102.
17. \* Ask the class to suggest how we could total all the Heads. Ask them to discover a function that will do this. Explain how the list of functions is divided into logical categories. When resolved, students go to Cell E3 and enter the formula**=COUNTIF(C3:C102,"Heads")**
18. In Cell E4, enter the formula **=COUNTIF(C3:C102,"Tails")**
19. In Cell E5, enter formula **=SUM(E3:E4)**
20. Finished. Press F9 to recalculate (ie toss 100 coins and total the outcomes!)

**Extension**

Ask students if we could create a button to automate the toss action. This is demonstrated in *Coin toss spreadsheet with macro (XLSM)*. First, a macro for F9 is created, then a button is drawn (available under Developer on the Excel ribbon controls), then it is assigned to a Form Button control by right-clicking > Assign Macro...

**Optional modification/enhancement**

An alternative formula is:
**=RANDBETWEEN(bottom integer, top integer)**
This generates an integer (no decimals!) that is between or includes the two given integers.

Then use:
**=IF(Cell=1, "Heads", "Tails")**

Students can also devise other problems that could be automated using a spreadsheet.

## Learning map and outcomes

* Students construct interactive spreadsheets designed to address particular needs.
* This lesson also demonstrates an approach to programming known as rapid application development (RAD).
* Students create a tool for solving a mathematical problem.
* You could also focus on the skillset and mindsets that learners mind need to adopt and use during this project, this ties in with the Creative and Critical Thinking Capabilities.

## Learning input

1. Introduce the 'Odds and evens' problem. State that the questions we want to answer are:
Does every number go to 1 or do some go on forever?
Is there a pattern?
2. Hand out *'Odds and evens'* worksheet.
3. Explain the rules:
Is it even? Then divide by 2.
Is it odd? Then multiply by 3 and add 1.
Is it 1? Then stop!
4. Students complete the worksheet until the realisation occurs that shortcuts are possible.
5. Students discuss the technique of using these shortcuts.
6. Explain that this problem is one of the great unsolved problems in mathematics.
Known as the 'Collatz conjecture', it states that no matter what number you start with, you will always eventually reach 1. The sequences are also known as 'hailstone numbers' because the values experience repeated descents and ascents like hailstones in a cloud.

Most mathematicians think the conjecture is true because experimental evidence supports it, but it has never been proven.

The longest chain for any starting number less than 100 million is 63,728,127, which has 949 steps!

## Learning construction

1. Pose the problem of how we might use a spreadsheet to speed up these calculations.
2. Supply the final working spreadsheet as a file: *Odds and evens spreadsheet – student example (XLS)*
3. Students follow the steps below to build a spreadsheet to simulate the 'Odds and evens' problem; however, this activity is intended as a series of problems for students to solve with your prompting where appropriate.

**Basic**

1. Fill Column A with consecutive integers 1 to 100.
2. Devise the main formula and place in Cell B1.
3. Fill formula across (suggest to Column DZ).
4. Select whole of Row 1 and fill down to Row 100.
5. Autofit columns to data content.

Modifications to enhance the tool

* Hide all the 0s (conditionally format these to white text colour).
* Extend rows up to 105, which will capture an interesting feature for chains for integers 99–103.
* Insert additional column at extreme left to record the length of each chain using Fill down with a suitable formula to count all non-zero entries in each row.
**=COUNTIF(B2:DZ2,">=1")**

This will give a count of the length of each chain.
* Insert suitable headings on the first two columns.
* Insert in top row a formula in a cell to record the length of the longest chain; eg **=MAX(A2:A100)**
* Students improve the interface of their spreadsheet.

**Challenge!**

Insert in top row a formula in a cell to record the integer with the longest chain from 1 to 105.

## Learning demo

Demonstrate other methods of iterative automation using the Python program supplied:
*Odds and evens* (PY)

A standalone app (both Windows and Macintosh apps provided) is also supplied to test for chain lengths:

* *Odds and evens.app* (Mac) (ZIP)
* *Odds\_and\_evens.exe* (Win)

Students use their spreadsheet to enter integers and find the longest chain lengths. Students may use the Python program provided or the standalone app to test for chain lengths. Discuss whether any patterns are evident in chain lengths.

## Curriculum links

| Links with Digital Technologies Curriculum Area |
| --- |
| **Strand** | **Content Description** |
| **Processes and Production Skills** | Analyse and visualise data interactively using a range of software, including spreadsheets and databases, to draw conclusions and make predictions by identifying trends and outliers (AC9TDI10P02).Model and query entities and their relationships using structured data (AC9TDI10P03). |

## Assessment

Note: Criteria are cumulative

|  | **Quantity of knowledge** | **Quality of understanding** |
| --- | --- | --- |
| **Automating spreadsheets** | No evidence of understanding | Student is able to use a prepared spreadsheet to operate an automated calculation | Student is able to create an automated calculation using a formula given to them | Student is able to create an automated calculation based on a formula they develop themselves | Student is able to generate an automated spreadsheet for a scenario they devise themselves |
| **Optional score** | 0 | 1 | 2 | 3 | 4 |