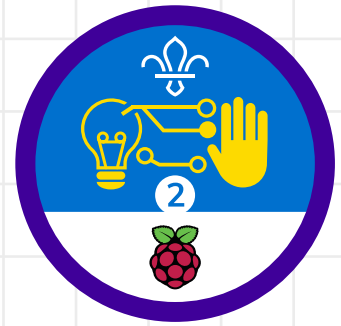# Leave the teams up to Scratch

Overview

This activity meets **requirement three** of **stage two** of the **Digital Maker Staged Activity Badge** ('Make a simple digital creation that uses code to interact with the wider world through inputs (such as buttons or typing on a keyboard) and outputs (such as a computer screen, sound, or lights)').

This activity shows people how to create a program in Scratch that asks a user to enter a list of names and how many teams they need. The program will then create random teams and display them on the screen.

60 minutes

Individuals or pairs

If you're running this activity without internet access, you'll need to download the software ahead of time. You'll probably also want to print handouts.

Wherever you have access to computers

You will need:

- ○ Laptops or desktop computers (one each person or pair)
- ○ Scratch editor (online: **rpf.io/scratch-new** or offline: **rpf.io/scratchoff**)
- ○ Activity handouts

If your meeting place has internet access

Run Scratch online in a web browser using this link: **rpf.io/scratchon**

If your meeting place doesn't have internet access

Download the Scratch desktop application before you begin. Check out the Scratch guide at **rpf.io/scouts-scratch** for instructions.

Key messages

- ○ People can talk to computers using external inputs from devices like keyboards.

- ○ People can make short programs that can perform tasks for us.

Scouts

Raspberry Pi

Leave the teams up to Scratch

Leader instructions

1 If people haven't used it before, introduce Scratch. Explain that they'll be coding using blocks that join together like Lego bricks. Explain that the blocks are colour-coded to make them easier to find.

2 Show how to write a very simple program (for example, you could program the Scratch cat to move or say something) and show everyone how to test it.

3 Everyone should think about when they might use their team chooser. For example, they could use it to pick groups for a game or activity.

Alternatives

- Create a similar program using different technology (for example, you could try using a micro:bit if it's available).

Safety

If young people are working online, they should ask for permission before viewing any other websites. It's a good idea to set up parental controls – you can find instructions on the NSPCC website (**rpf.io/scouts-nspcc-online**). You should also give each young person a 'Stay safe' leaflet (**rpf.io/scouts-staysafe**).

Community and sharing

- You can share Scratch projects on the Scratch website by logging in and using the 'Share' option.

- You can share your program with younger groups who need help picking teams but haven't started to learn any coding.

Adaptability

- You can learn more about using sprites in Scratch by experimenting with the Scratch cat. For example, you could try to make it move or point when it calls out the names.

Scouts

Raspberry Pi

Leave the teams up to Scratch

You're going to use Scratch to make a computer program that asks the user to enter a list of names and how many teams they need. The program will then randomly divide the names into the desired number of teams. This will be useful for splitting a group into smaller teams for games and activities.

Part one: get input from the keyboard

First, you'll create a new Scratch project and make it ask the user for their list of names and desired number of teams.

1 When you open Scratch, make sure that you're using a new project. To do this, click on 'File', then on 'New'.

2 You'll need a `variable` and a `list` to store the number of teams and the list of names. To create a new variable, go to the 'Variables' menu and click on 'Make a Variable'. This variable will be for **all sprites**. Name it 'Number of teams' and click on 'OK'.

3 To create a new list, go to the 'Variables' menu and click on 'Make a List'. Give it a name (for example, 'List of names'), then click on 'OK'. The list should also be for all sprites.

Did you know

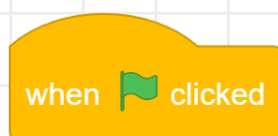The very first version of Scratch was created in 2003.

Scouts

Raspberry Pi
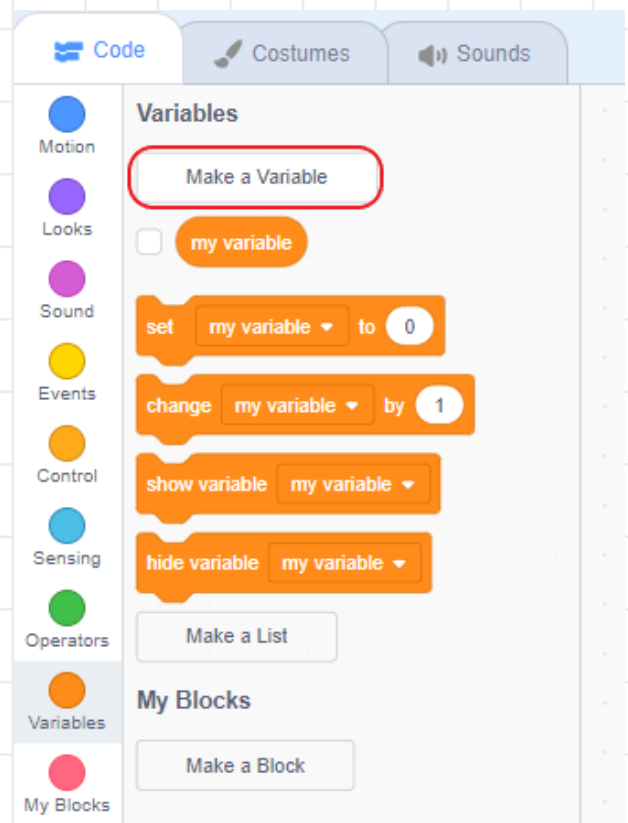
Leave the teams up to Scratch

4   Add a `when green flag clicked` block to the program from the 'Events' menu.

```
when 🚩 clicked
```

5   Next, you'll ask the user how many teams they want the program to make. Add an `ask and wait` block beneath the `when green flag clicked` block.
You can find the block in the 'Sensing' menu. Enter the question 'How many teams would you like?' into the blank space in the block.

```
when 🚩 clicked
ask  How many teams would you like?  and wait
```
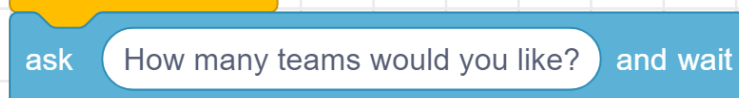
6   To put the user's answer into the variable `Number of teams`, add a `set my variable to` block beneath the `ask and wait` block. You can find this block in the 'Variables' menu. Use the drop-down menu in the block to change `my variable` to `Number of teams`. Finally, add an `answer` block into the space in the `set to` block. You'll find the `answer` block in the 'Sensing' menu. The `answer` block will store the answer that the user gives to the question in the `ask and wait` block.

```
when 🚩 clicked
ask  How many teams would you like?  and wait
set  Number of teams ▼  to  answer
```
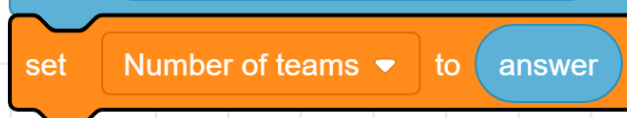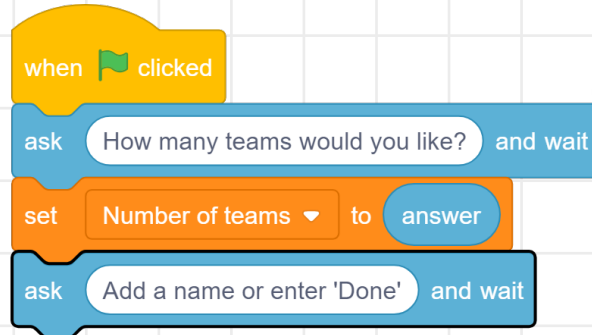
7   It's important to test your code often. Click on the green flag and the Scratch cat should ask you the question 'How many teams would you like?'. Enter a number, press **Enter**, and the number should appear in the box on the Stage that shows the variable (on the top left-hand side).

Scouts   Raspberry Pi

Leave the teams up to Scratch

8 Now you need to ask for the list of names. First, add an `ask and wait` block beneath the `set` block and type in the instruction 'Add a name or enter 'Done''. This will ask for the first name.

```
when [flag] clicked
ask (How many teams would you like?) and wait
set [Number of teams ▼] to (answer)
ask (Add a name or enter 'Done') and wait
```

9 Add a `repeat until` block from the 'Control' menu beneath your `ask and wait` block.

10 Select an '=' block from the 'Operators' menu. Into this, you can place an `answer` block on the left and the word 'Done' on the right.

```
(answer) = (Done)
```

11 Use the '=' operator block to keep checking if the user enters the word **Done**. Inside your loop, add an `add to list` block from the 'Variables' menu and another `ask and wait` block to repeat the instruction. Your code should look like this:

```
when [flag] clicked
ask (How many teams would you like?) and wait
set [Number of teams ▼] to (answer)
ask (Add a name or enter 'Done') and wait
repeat until ((answer) = (Done))
    add (answer) to [List of names ▼]
    ask (Add a name or enter 'Done') and wait
```

Scouts   Raspberry Pi

Leave the teams up to Scratch
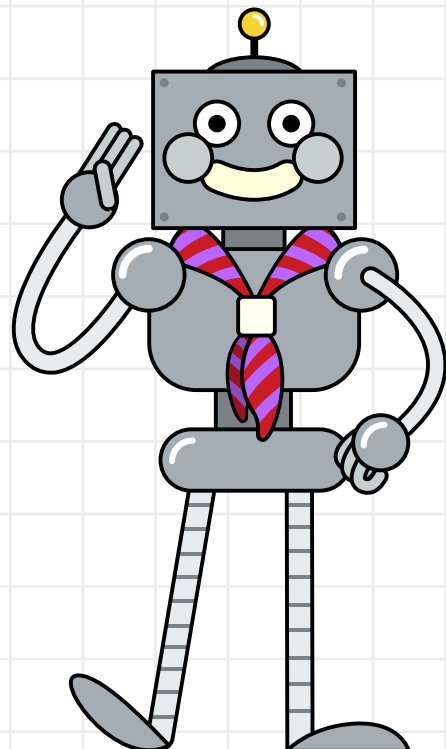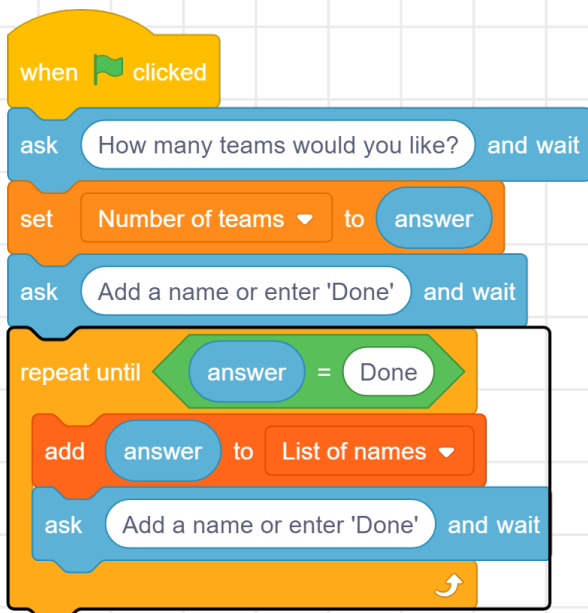
12 Test your code once more. Can you see your list fill up? If you run your code a few times, you'll see that the list keeps the old names. To prevent this, you can **delete all** of your list. Add the block from the 'Variables' menu to the start of your code, directly below the **when green flag clicked** block.

```
when [flag] clicked
delete all of  List of names ▾
ask  How many teams would you like?  and wait
set  Number of teams ▾  to  answer
ask  Add a name or enter 'Done'  and wait
repeat until  < answer = Done >
    add  answer  to  List of names ▾
    ask  Add a name or enter 'Done'  and wait
```
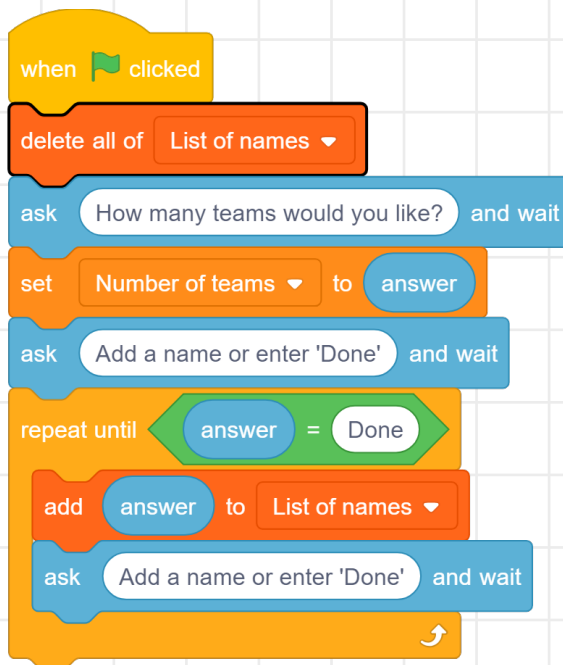
13 To make it easier to test the program, you can use **add** blocks to fill the list automatically. Keep the code that you've already written to the side for later.

```
when [flag] clicked
delete all of  List of names ▾
set  Number of teams ▾  to  3
add  Jane  to  List of names ▾
add  Bill  to  List of names ▾
add  Bob  to  List of names ▾
add  Rebecca  to  List of names ▾
add  Lucy  to  List of names ▾
add  Robert  to  List of names ▾
```

```
ask  How many teams?  and wait
set  Number of teams ▾  to  answer
ask  Add a name or enter 'Done'  and wait
repeat until  < answer = Done >
    add  answer  to  List of names ▾
    ask  Add a name or enter 'Done'  and wait
```

Scouts

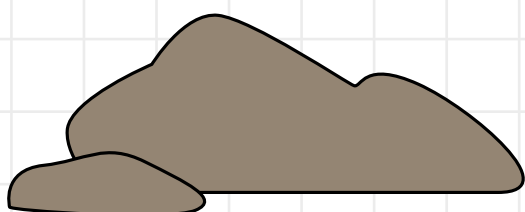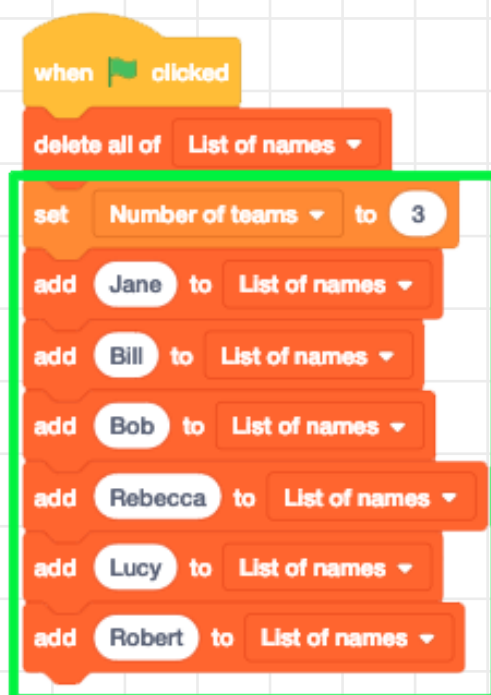Raspberry Pi

Leave the teams up to Scratch

Part two: display the teams

Now you'll go through your list and assign a random name to a team. For each name, you'll change the team number to keep the number of names per team fair. Then, the Scratch cat will say each name and the team that the person is in.

1 Create two new variables called **Team** (to store the current team) and **Next team member** (to store the next team member).

2 Add a **set to** block from the 'Variables' menu to the bottom of your code and set the variable **Team** to 1.

```
when 🏴 clicked
delete all of  List of names ▼
set  Number of teams ▼  to  3
add  Jane  to  List of names ▼
add  Bill  to  List of names ▼
add  Bob  to  List of names ▼
add  Rebecca  to  List of names ▼
add  Lucy  to  List of names ▼
add  Robert  to  List of names ▼
```

3 Add a **repeat until** block to repeat until the length of the list is equal to 0.

```
add  Lucy  to  List of names ▼
add  Robert  to  List of names ▼
set  Team ▼  to  1
repeat until ⟨ length of  List of names ▼  =  0 ⟩

```

Scouts   Raspberry Pi
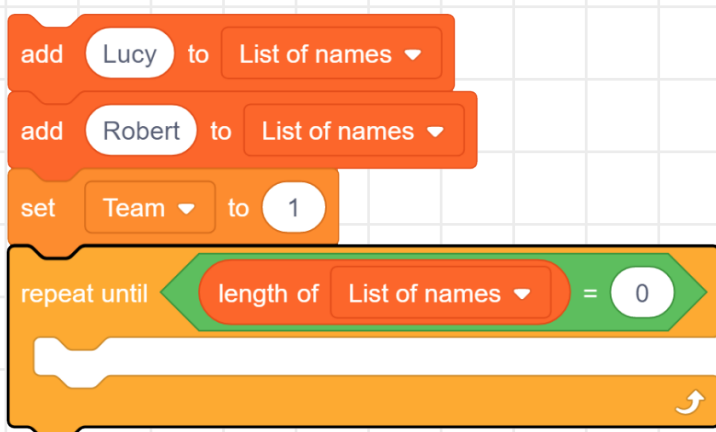
Leave the teams up to Scratch

4 Use **set to** and `pick random` blocks from the 'Variables' and 'Operators' menus to set the **Next team member** variable to a random number. Add a **length of** block from the 'Variables' menu, so that the program will pick a random number between one and the number of names.

```
add (Lucy) to (List of names ▼)
add (Robert) to (List of names ▼)
set (Team ▼) to (1)
repeat until < (length of (List of names ▼)) = (0) >
    set (Next team member ▼) to (pick random (1) to (length of (List of names ▼)))
```
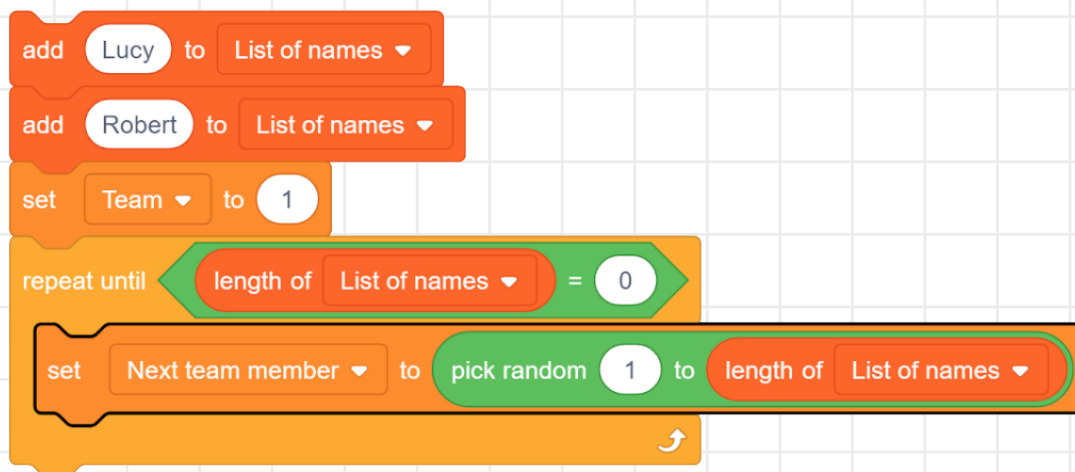
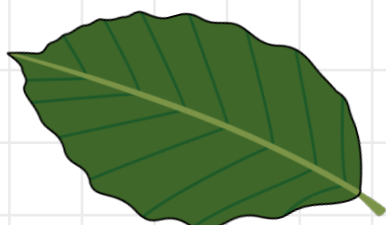**5** Add a `say for 2 seconds` block from the 'Looks' menu inside the `repeat until` block to assign the **Next team member** name in your list to the current team. You'll need to use `join` blocks for this – see the image for help.

```
add (Lucy) to (List of names ▼)
add (Robert) to (List of names ▼)
set (Team ▼) to (1)
repeat until < (length of (List of names ▼)) = (0) >
    set (Next team member ▼) to (pick random (1) to (length of (List of names ▼)))
    say (join (item (Next team member) of (List of names ▼)) (join (is in Team:) (Team))) for (2) seconds
```

Scouts

Raspberry Pi
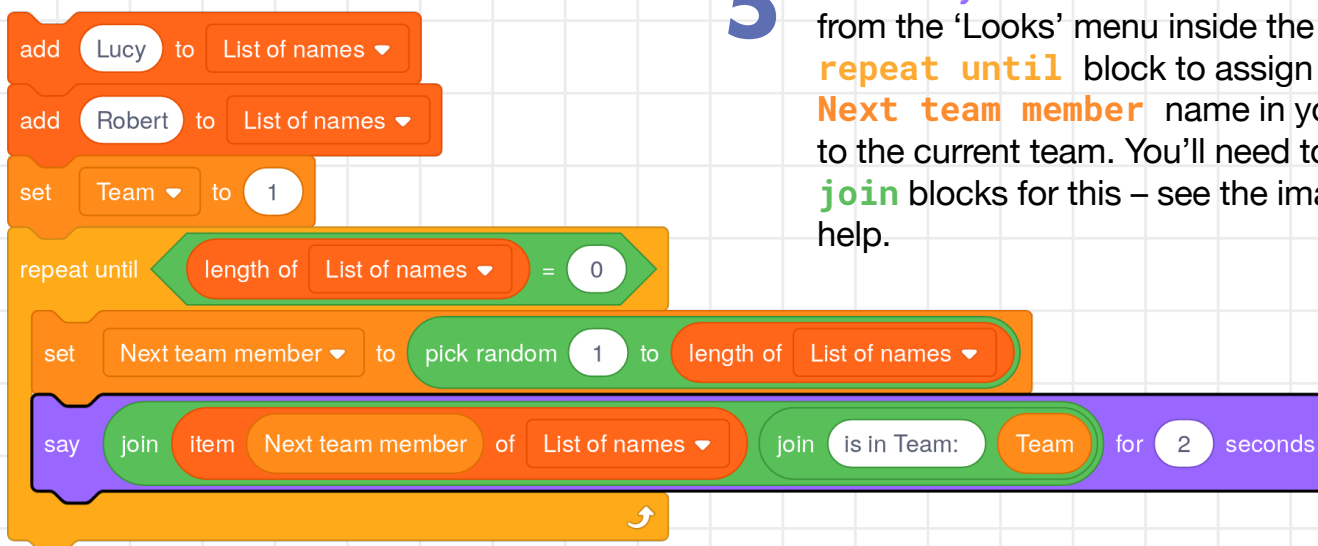
Leave the teams up to Scratch
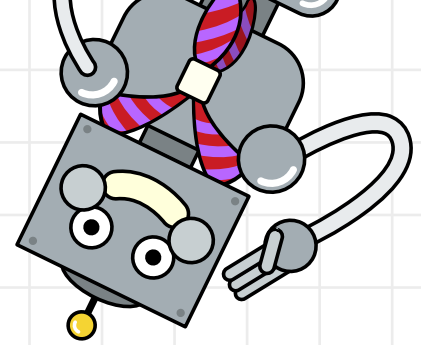
6 Add a **delete** block to delete that name from the list and add a **change by** block to increase the **Team** variable by one.

```
add ( Lucy ) to [ List of names ▼ ]
add ( Robert ) to [ List of names ▼ ]
set [ Team ▼ ] to ( 1 )
repeat until < ( length of [ List of names ▼ ] ) = ( 0 ) >
    set [ Next team member ▼ ] to ( pick random ( 1 ) to ( length of [ List of names ▼ ] ) )
    say ( join ( item ( Next team member ) of [ List of names ▼ ] ) ( join ( is in Team: ) ( Team ) ) ) for ( 2 ) seconds
    delete ( Next team member ) of [ List of names ▼ ]
    change [ Team ▼ ] by ( 1 )
```
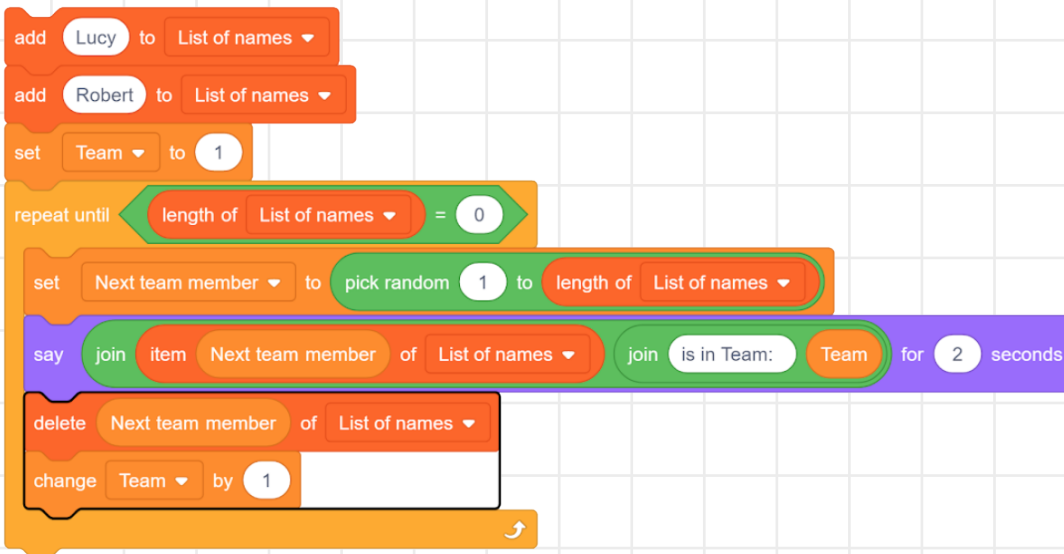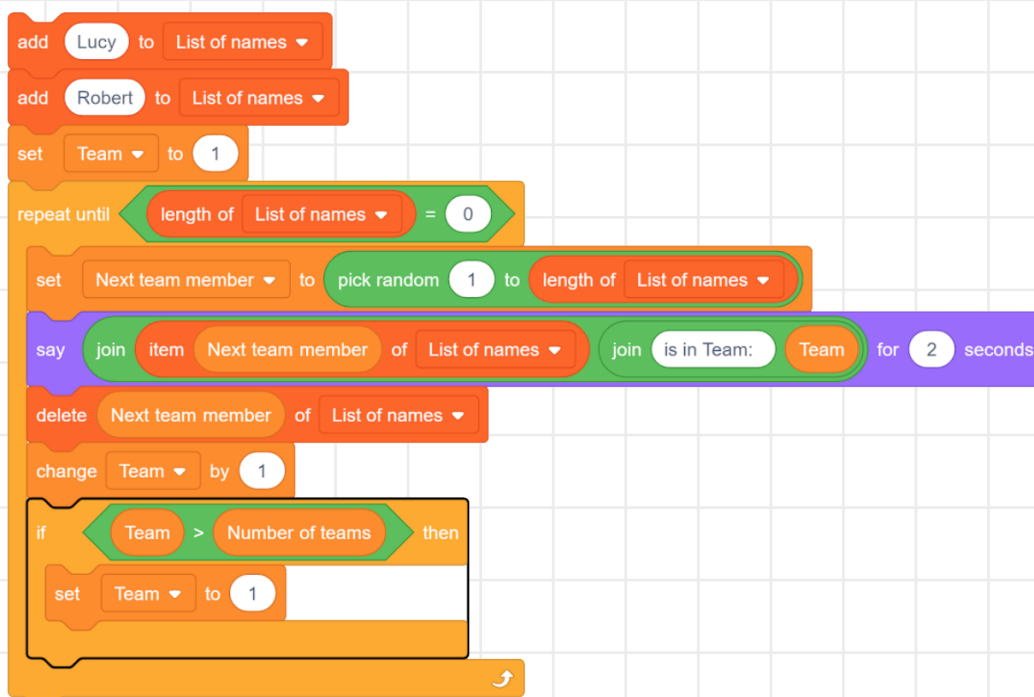
7 You need to make sure that the **Team** variable never goes above the **Number of teams** that the user has entered. Add an **if then** block from the 'Control' menu to check for this and a **set to** block to set **Team** back to one if it happens.

```
add ( Lucy ) to [ List of names ▼ ]
add ( Robert ) to [ List of names ▼ ]
set [ Team ▼ ] to ( 1 )
repeat until < ( length of [ List of names ▼ ] ) = ( 0 ) >
    set [ Next team member ▼ ] to ( pick random ( 1 ) to ( length of [ List of names ▼ ] ) )
    say ( join ( item ( Next team member ) of [ List of names ▼ ] ) ( join ( is in Team: ) ( Team ) ) ) for ( 2 ) seconds
    delete ( Next team member ) of [ List of names ▼ ]
    change [ Team ▼ ] by ( 1 )
    if < ( Team ) > ( Number of teams ) > then
        set [ Team ▼ ] to ( 1 )
```

Scouts   Raspberry Pi

Leave the teams up to Scratch

8 Test your code to make sure that everything works. You should see the Scratch cat announce each name with a team number beside it and see the list slowly empty as each name is displayed. If your code is working as expected, you can add your code from part one back in.

```
when 🏳 clicked
delete all of   List of names ▾
set   Number of teams ▾  to  3
add  Jane  to  List of names ▾
add  Bill  to  List of names ▾
add  Bob  to  List of names ▾
add  Rebecca  to  List of names ▾
add  Lucy  to  List of names ▾
add  Robert  to  List of names ▾
```

```
ask  How many teams?  and wait
set  Number of teams ▾  to  answer
ask  Add a name or enter 'Done'  and wait
repeat until   answer  =  Done
    add  answer  to  List of names ▾
    ask  Add a name or enter 'Done'  and wait
```

```
Number of teams    3
List of names
1  Bob
2  Lacy
3  Miranda
4  Tim
5  Dave
6  Ronald
+    length 6    =
```

Miranda is in Team: 3

Scouts

Raspberry Pi

Leave the teams up to Scratch

Discuss

What types of games or activities could you use this program for? Think about what it means to be a good team member. Ask the young people to think about the different types of teams that they're part of – they could think about sports teams, school teams, or other teams.

Tip

If you need a block that you've already used, you can duplicate it instead of searching for it again by right-clicking on the block.

Scouts

Raspberry Pi