

# Numbers



## Unit 1 – Application – Teacher

7 8 9 10 11 12



TI-Nspire™



Coding



Language



Student



50 min

## Introduction

A wonderful British game show called: “Countdown” began in 1982. Almost thirty years later, SBS (Australia) aired a similar show titled: “Letters and Numbers”. In the “Numbers” round, a contestant nominates the proportion of large and small numbers, a total of six, from a mix of large: {25, 50, 75, 100} and small {1, 2, 3, ... 10}. A random three-digit target number is then generated. Players have 30 seconds to create an expression using a combination of the four operations: + - ÷ × and parenthesis as applicable that is as close to the target number as possible. Each number can be used once, at most, however all six numbers did not have to be used. Players were awarded points according to their proximity to the target number.

**Example:** Suppose the contestant chose three large and three small numbers. The randomly generated selection might consist of the following: {25, 50, 100, 2, 3, 7} Target number: 72 (Randomly generated)

One solution might be:  $25 \times 50 \div 2 + 100 + 7 - 3 = 729$  This solution uses all 6 numbers.

Another option might be:  $3 \times (2 \times 100 + 50 - 7) = 729$  This solution uses only 5 of the 6 numbers.

Both results achieve a perfect score. Fewer points are obtained for numbers not equal to 729, based on proximity.

Like all games, the opportunity to practice means greater skills. It is easy to write a program that will generate samples!



It is assumed that you have completed:

### Unit 1 Assigning Values to Variables - Skill Builder 1

You may return to the Skill Builder exercise at any time to review the instructions.



## Numbers

Start a new document and create a program titled:

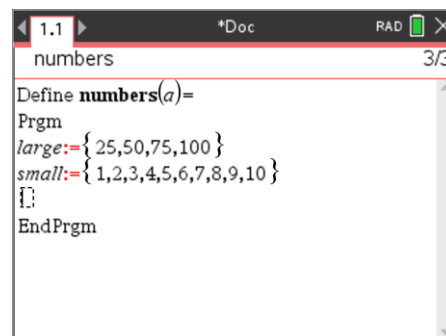
Numbers

Enter the variable ‘a’ to represent the quantity of large numbers.

Two lists of numbers need to be created:

Large:={25, 50, 75, 100}

Small:={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}



Three more lines of code need to be added.

```
Disp randSamp(large,a)
```

```
Disp randSamp(small,6 – a)
```

```
Disp randInt(100,999)
```

Once these lines of code have been included check the syntax, store the program and run it by pressing: (Ctrl + R).

Enter a value for the quantity of large numbers. Example: numbers(2)

### Question: 1.

Run the numbers program several times to practice. Record the output and your attempts to solve each problem then answer the following:

- a. How does the program know how many small numbers to generate?

**Answer:** Large = a, small = 6 – a, which is the amount left over after large have been selected.

- b. Are the large numbers ever repeated?

**Answer:** Yes. **Note:** It is possible that students may not see a duplicate.

- c. What does the command: randSamp() do?

**Answer:** It randomly selects the specified quantity of elements from the list. randSamp(list,qty)

A small change has been made to the **numbers** program.

Read through your code and the code shown opposite.

Adjust your program accordingly and then run the revised program several times.

### Question: 2.

What change was made and how did it affect the outcome?

**Answer:** randSamp(large,n,1). The one in the syntax here means there are no repeats in the random selection.

Some additional text commands and a new command "Wait" have been included in the program shown opposite.

**Menu** > **Control** > **Wait**

Include the additional 'display' commands, check the syntax, store and run the program.

### Question: 3.

What is the role of the "Wait" command?

**Answer:** Program execution is halted for the specified amount of time.

### Question: 4.

The game could be made a little easier if contestants could access seven numbers in order to complete the challenge. Which section of the program would need to change in order to accommodate this option?

**Answer:** One change only: randsamp( small, 7 – n)