Linear search: appropriate data

Practice 1

Whilst a linear search is not the most efficient algorithm, it is the most appropriate in some situations. The options below show examples of lists of data. The actual lists would be far longer but the nature of each list would be as shown in the example.

From the choices below, select the list for which it would be most appropriate to use a **linear search** if you need to find something in the list.

- ["Yellow", "Teal", "Purple", "Green", "Blue"]
- [393, 783, 656, 143, 727]
- [1, 38, 41, 56, 64, 78]
- ["TN-0094", "TN-0184", "TN-0412", "TN-0905", "TN-0939"]

Linear search: order steps

Practice 2

Amaia coaches a school basketball team. She keeps the scores that the team got in the latest tournament in a list called `basketball_finals`. You can see the items of the list below:

basketball_finals							
120	250	101	150	80	95	147	165

Part A

Amaia is using the linear search algorithm to find out if the team scored 200 points in any of the games. How many comparisons will the linear search algorithm perform before it finishes?

Part B

Amaia is using the linear search algorithm to find out if the team scored 150 points in any of the games. Put in order the steps that the linear search algorithm will follow when searching for the item `150` in the list `basketball_finals`.

Available items

- Go to the next item in the list: 250
- Compare 250 to 150: 250 is not equal to 150
- Compare 101 to 150: 101 is not equal to 150
- The item was found, stop searching
- Compare 120 to 150: 120 is not equal to 150
- Start from the first item in the list: 120
- Compare 150 to 150: 150 is equal to 150
- Go to the next item in the list: 101
- Go to the next item in the list: 150

Quiz:
[STEM SMART Computer Science Week 7]

Linear search: use of flag

Challenge 1

Consider the pseudocode implementation below:

Pseudocode

```
FUNCTION linear_search(data_set, item_sought)
    index = -1
    i = 0
    found = False
    WHILE i < LEN(data_set) AND found == False
        IF data_set[i] == item_sought THEN
            index = i
            found = True
        ENDIF
        i = i + 1
    ENDWHILE
    RETURN index
ENDFUNCTION
```

Part A

Linear search can be a very inefficient algorithm because, in some versions, it iterates through the whole of the list even when the item has already found.

In this more efficient version of the algorithm, which variable is used to terminate the loop when the item has been found? Please write the variable's name and nothing else.

Part B

Consider the following list:

```
data = [262, 764, 301, 220, 471, 900, 139, 779, 439, 190]
```

What would be returned from the following statement?

```
linear_search(data, 471)
```

Quiz:
STEM SMART Computer Science Week 7

Binary search: appropriate data

Practice 1

You have been provided with some cards that are in the following order:

Why could you **not** use a binary search for this data?

- There is not enough data.
- The data is not sorted in an ascending (low to high) or descending (high to low) order.
- The data is not a list of numbers.

Quiz:
STEM SMART Computer Science Week 7

Binary search: max comparisons 1

Practice 2

What is the **maximum** number of comparisons that a binary search will apply for an array of length 30.

Quiz:
STEM SMART Computer Science Week 7

Binary search: max comparisons 2

Practice 2

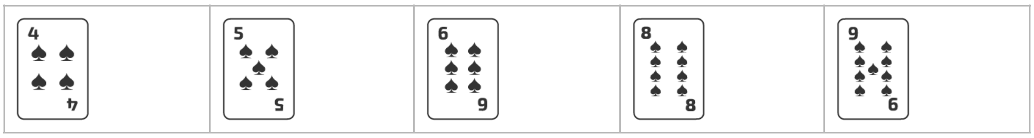
A warehouse has the details of $10,000$ products stored in a list, **ordered by product number**. What is the **maximum number of comparisons** that will be made to find a specific product, by product number, using a **binary search**?

Quiz:
STEM SMART Computer Science Week 7

Binary search: midpoint

Practice 1

You have been given a set of cards that are in the following order:



A binary search is going to be used to search for an item in the list. Which value card is at the first midpoint calculated by the algorithm?

Quiz:
STEM SMART Computer Science Week 7

Binary search: complete algorithm

Challenge 1

Below is a function for carrying out a binary search expressed in pseudocode.

The statement to calculate the midpoint is incomplete. Using the values below, **complete the missing part of the statement** to determine the midpoint.

```
FUNCTION binary_search(data_set, item_sought)
    index = -1
    found = False
    first = 0
    last = LEN(data_set) - 1
    WHILE first <= last AND found == False
        midpoint = [        ][        ][        ]
        IF data_set[midpoint] == item_sought THEN
            index = midpoint
            found = True
        ELSE
            IF data_set[midpoint] < item_sought THEN
                last = midpoint - 1
            ELSE
                first = midpoint + 1
            ENDIF
        ENDIF
    ENDWHILE
    RETURN index
ENDFUNCTION
```

Items:

(first + last) | 2 | / | last | DIV | LEN(data_set) | first

Quiz:
STEM SMART Computer Science Week 7

Binary search: order steps 1

Practice 2

Tom sells knitwear via his website. He has a list with the number of products that he's sold in the last ten days.

products_sold									
11	16	17	22	25	30	35	46	57	72

Tom is using the binary search algorithm to search through the list of sales.

The binary search implementation uses the following formula to calculate the midpoint position:

```
midpoint = (first + last) DIV 2
```

Remember that integer or floor division (DIV) is when any remainder of a division is discarded. For example, $11 \div 5 = 2$ remainder 1, and so 11 DIV 5 will return 2.

Part A

Tom wants to find out if there is a day that he sold 22 products.

Fill in the blanks with the values provided below to complete the steps that a binary search will follow when searching for 22 in the `products_sold` list. The list of items has been copied below to help answer the question.

`products_sold`									
11	16	17	22	25	30	35	46	57	72

Take the ordered list `products_sold` and the search item 22.
Initially, set the range of items where the search item might be found to be the entire list.

Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is greater than 22.
Change the range to focus on the items before the midpoint.

Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is less than 22.
Change the range to focus on the items after the midpoint.

Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is ______ 22.
Change the range to focus on the items ______ the midpoint.

Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is equal to 22.

The search item was found in the list, stop searching.

Items:

less than | greater than | before | after | 11 | 16 | 17 | 22 | 25 | 30 | 35 | 46 | 57 | 72

Part B

Tom wants to find out if there is a day that he sold 35 products.

Fill in the blanks with the values provided below to complete the steps that a binary search will follow when searching for `35` in the `products_sold` list. The list of items has been copied below to help answer the question.

products_sold									
11	16	17	22	25	30	35	46	57	72

Take the ordered list `products_sold` and the search item `35`.
Initially, set the range of items where the search item might be found to be the entire list.
Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is less than `35`.
Change the range to focus on the items after the midpoint.

Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is ______ `35`.
Change the range to focus on the items ______ the midpoint.

Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is ______ `35`.
Change the range to focus on the items ______ the midpoint.

Find the item at the midpoint position: ______.
Compare the item at the midpoint to the search item: the item at the midpoint is equal to `35`.
The search item was found in the list, stop searching.

Items:

less than | greater than | before | after | 11 | 16 | 17 | 22 | 25 | 30 | 35 | 46 | 57 | 72

Quiz:
STEM SMART Computer Science Week 7

Binary search: trace

Challenge 2

The following list of items is stored in an array:

items							
index	0	1	2	3	4	5	6
value	2	4	5	8	12	15	18

The following algorithm has been coded and will be used to search for the number 12 as highlighted in the array.

Pseudocode

```
FUNCTION binary_search(items, search_item)
    found = False
    found_index = -1
    first = 0
    last = LEN(items) - 1

    WHILE first <= last AND found == False
        midpoint = (first + last) DIV 2
        IF items[midpoint] == search_item THEN
            found_index = midpoint
            found = True
        ELSEIF items[midpoint] < search_item THEN
            first = midpoint + 1
        ELSE
            last = midpoint - 1
        ENDIF
    ENDWHILE

    RETURN found_index
ENDFUNCTION
```

Complete the trace table for the algorithm. The first pass has already been filled for you.

first	last	midpoint	items [midpoint]	found_ index	found
0	6	3	8	−1	False
					False
				4	True