Summative assessment – Questions

1. Read the description below and then select which term it is referring to:

“The process of removing unnecessary information and focussing on the important details, allowing you to develop a general idea of what the problem is and how to solve it.”

1. Decomposition
2. Abstraction
3. Algorithmic thinking
4. Computational thinking
5. A sample of alphabetically sorted data is shown below:

| Crane | Finch | Heron | Kiwi | Owl | Stork | Wren |
| --- | --- | --- | --- | --- | --- | --- |

1. Which searching algorithm would result in the least number of comparisons when the search item is “Crane”?

|  |
| --- |

1. Which searching algorithm would result in the least number of comparisons when the search item is “Wren”?

|  |
| --- |

1. **List** each of the items that would be compared to the search item “Robin” when performing a binary search.

|  |
| --- |

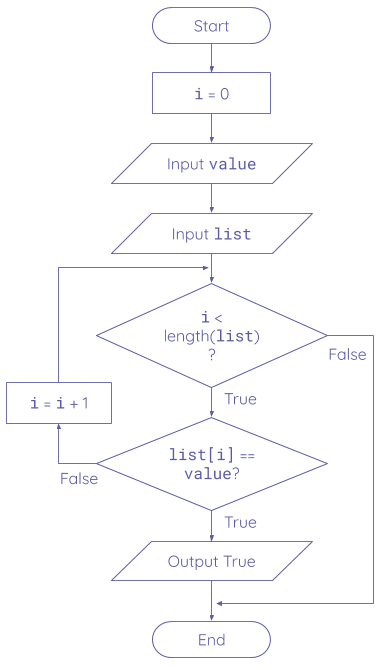
1. Read the description below and then select which algorithm it is referring to:

“Split data until each item is in a list of its own and then combine pairs of lists repeatedly so that the items are in order.”

1. Bubble sort
2. Merge sort
3. Insertion sort
4. Binary search
5. Read the description below and then select which algorithm it is referring to:

“Move through a list repeatedly, comparing items next to each other and swapping them if they are in the wrong order.”

1. Bubble sort
2. Merge sort
3. Insertion sort
4. Linear search
5. Analyse the flowchart below. Which algorithm does this represent?



1. Bubble sort
2. Merge sort
3. Linear search
4. Binary search
5. A sample of data is shown below:

| Frozen | Cats | Aladdin | Moana | Grease | Annie |
| --- | --- | --- | --- | --- | --- |

1. What order will the data be after the first pass of a bubble sort?
   1. Cats, Frozen, Aladdin, Moana, Annie, Grease
   2. Cats, Frozen, Aladdin, Moana, Grease, Annie
   3. Cats, Aladdin, Frozen, Grease, Annie, Moana
   4. Cats, Aladdin, Frozen, Annie, Grease, Moana
2. What order will the data be after the third pass of a bubble sort?
   1. Aladdin, Cats, Frozen, Moana, Grease, Annie
   2. Aladdin, Cats, Annie, Frozen, Grease, Moana
   3. Aladdin, Annie, Cats, Frozen, Grease, Moana
   4. Aladdin, Annie, Cats, Grease, Frozen, Moana
3. A sample of data is shown below:

| Dalí | Matisse | Sanzio | Picasso | Munch | Renoir |
| --- | --- | --- | --- | --- | --- |

1. When applying a merge sort to this data, which items would be in the two groups after the split?

| **Group 1** |  |
| --- | --- |
| **Group 2** |  |

1. What would the groups contain after a second split?

| **Group 1** |  |
| --- | --- |
| **Group 2** |  |
| **Group 3** |  |
| **Group 4** |  |

1. Select all the statements that are true:

* Linear search can only be performed on unsorted data
* Binary search can only be performed on sorted data
* Linear search uses floor division to find the midpoint of the list
* Bubble sort is always more efficient than insertion sort
* Merge sort is a divide and conquer algorithm

1. Read the Python program below:

| 1  2  3  4  5  6  7  8  9  10 | def an\_algorithm(items):  num\_items = len(items)  i = 1  while i < num\_items:  for current in range(num\_items - 1):  if items[current] > items[current+1]:  temp = items[current]  items[current] = items[current+1]  items[current+1] = temp  i = i + 1 |
| --- | --- |

1. **Complete** the trace table below only for lines 7-9 of the algorithm. The first line in the trace table contains the values for the current variable and the items list.

|  |  |  | items | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Line | current | temp | [0] | [1] | [2] | [3] |
|  | 0 | - | Pakistan | China | Greece | Chad |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |

1. What algorithm is this?

|  |
| --- |

1. If items is a list of 10 elements, how many comparisons would be made on line 6 during the first iteration of the outer while loop?

|  |
| --- |

1. The number of comparisons is currently the same during each pass of the inner for loop on line 5. One improvement to the algorithm would be to reduce the number of comparisons made during each pass.

**Explain** how you would implement this improvement in the Python program.

|  |
| --- |

Resources are updated regularly — the latest version is available at: [ncce.io/tcc](http://ncce.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](http://ncce.io/ogl).