Dynamic data structure: advantage

Practice 2

Which of these is an **advantage** of a dynamic data structure?

- ◯ It allocates memory to pointers to keep track of where things are

- ◯ It will overflow if it exceeds its allocated memory

- ◯ It makes efficient use of memory, using only as much memory as it needs

- ◯ The memory allocation is fixed so there is no need to check size when adding and removing data items

Raspberry Pi
Foundation

Dynamic vs static structures 2

Practice 2

Data structures can be static or dynamic. Which three of the statements are **correct**?

☐ The elements of a dynamic linked list can be accessed directly by index.

☐ A static array cannot be resized at runtime.

☐ A dynamic data structure is a collection of data in memory that can grow or shrink in size at runtime.

☐ A static array uses an allocation of contiguous (one after another) locations in main memory.

Quiz:

STEM SMART Computer Science Week 6

Array: 2D indexing

Practice 2

The table below illustrates the contents of a two-dimensional array that has been set up to store words for an application to help schoolchildren to learn their spellings.

Index	0	1	2	3
0	school	pull	where	here
1	path	floor	sugar	bread
2	accident	answer	eight	critical

When implemented in the program, the name of the array is `spelling_words`. The first index references the row number and the second index references the column number.

What will be displayed when the following line of code (shown in pseudocode) is run?

`PRINT(spelling_words[2,1])`

○ answer

○ path

○ sugar

○ pull

Quiz:
STEM SMART Computer Science Week 6

Array: trace pseudocode 1

Challenge 1

Emily is organising a charity event at her school. She needs to select four students to meet representatives from the charity when they come to visit.

She starts by creating an array `students` to hold the names of the eight students who raised money for the charity:

After she has added all the names, the array (in pseudocode) looks like this:

`["Miguel", "Laura", "Mariam", "Arthur", "Musa", "Magda", "Ben", "Diane"]`

Now Emily writes a program (expressed in pseudocode below) to pick four students:

Pseudocode

```
1  PROCEDURE pick_students()
2    students = ["Miguel", "Laura", "Mariam", "Arthur", "Musa", "Magda", "B
3    selected = 0
4
5    WHILE selected < 4
6      picked = RANDOM_INT(0,7)  // Return a random integer between 0 and 7
7      IF students[picked] != "Selected" THEN
8          PRINT(students[picked])
9          students[picked] = "Selected"
10         selected = selected + 1
11     ENDIF
12   ENDWHILE
13 ENDPROCEDURE
```

When the program is run, the sequence of random numbers is:

4, 1, 7, 1, 3

What is the output of the program?

○ Musa Laura Diane Arthur

○ Musa Laura Diane Selected Arthur

○ Musa Laura Diane Laura Arthur

○ Arthur Miguel Ben Mariam

Linked list: add a node to an unordered list Practice 2

Matthew has designed a system that keeps a list of films he wants to watch. When he hears about an interesting new film, he adds it to the list. When he has watched a film, he deletes it from the list.

The list of films is stored within the system as a linked list:

- Each node has a distinct memory location, which is shown in **Figure 1** as the number above the node
- Each node contains data (in this case, the name of the film) and a pointer to the memory location of the next node
- The node labelled head indicates the first element in the linked list (the **head** of the list)
- For the last element in the list, the next node pointer always points to a null value to mark the end of the list
- The list is **unordered** so when a new node is added, it is added to the **head** of the list

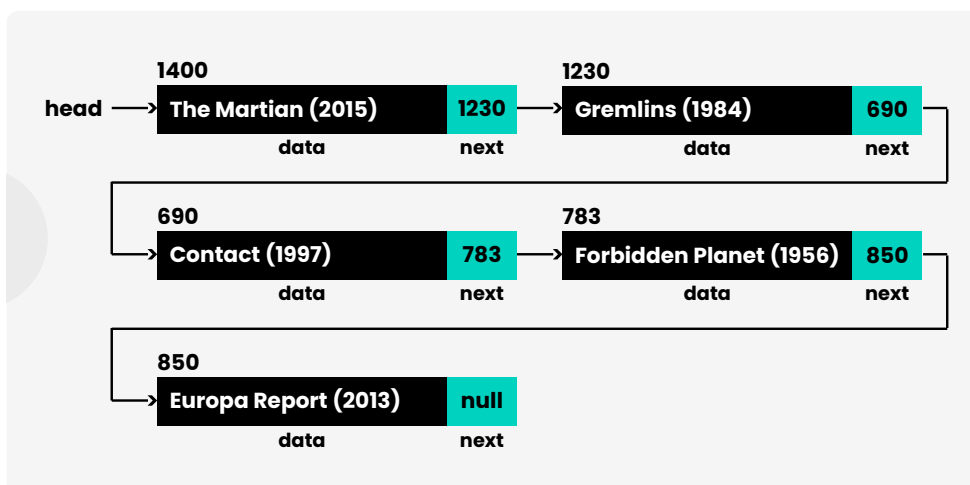The current state of the linked list is shown in **Figure 1** below:

Figure 1: The current state of the linked list

Matthew's friend recommends a new film called "Spiderman: Across the Spider-verse (2023)", and Matthew uses the system to add it to his list of films to watch.

A new node is created at memory location 250 to store the details of the film. In this new node, the value of data will be set to "Spiderman: Across the Spider-verse (2023)". What value will next be set to?

Linked list: add a node to an ordered list

Practice 2

Eli uses a system that keeps a list of the different birds he sees in his garden. The list of birds is stored within the system as a linked list:

- Each node has a distinct memory location, which is shown in **Figure 1** as the number above the node.
- Each node contains data (in this case, the name of the bird) and a pointer to the memory location of the next node.
- The node labelled head indicates the first element in the linked list (the **head** of the list).
- For the last element of the list, the next node pointer always points to a null value to mark the end of the list.
- The list is **ordered alphabetically (A to Z)**. When a new node is added, it is added **into the correct position** within the list.

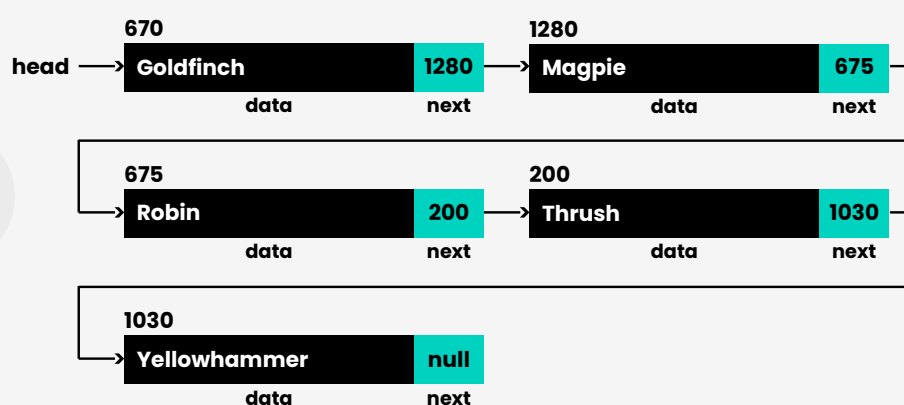The current state of the linked list is shown in **Figure 1** below:

Figure 1: The current state of the linked list

Eli spots a pigeon in the garden and records it in the system.

A new node is created at memory location 950 to store the details of the bird. In this new node, the value of data will be set to "Pigeon". What value will next be set to?

Quiz:
STEM SMART Computer Science Week 6

Array of records: advantages

Practice 1

Rudi has made a list of gifts that he would like to buy for his family to celebrate the festive break. He has written a program to help him keep track of his ideas. For each gift idea he wants to store:

- Name of the person (the gift is for)
- Description of the gift
- Likely cost
- Purchased (yes/no)

He has decided that an **array of records** is a suitable data structure for his program.

Why has Rudi decided that a record structure is a suitable container for each gift idea?

○ It can hold data of different data types.

○ There is no limit to the number of records he can store.

○ The elements of the record (the fields) can be referred to using an index number.

Quiz:

STEM SMART Computer Science Week 6

Raspberry Pi Foundation

Record: access 2

Challenge 1

Sergiu wants to create a system to catalogue his music collection. To get started, he has defined:

- a record structure - `MusicRecord` - for storing the information about a specific track
- an array of records - `album` - to store the details for all of the tracks on a specific album

The declarations for the record structure and the array are shown below:

Pseudocode

```
1  MusicRecord = RECORD
2      track_id: Integer
3      track_name: String
4      artist: String
5      duration: Time
6  ENDRECORD
7
8  ARRAY album[15]
```

Assuming that the tracks will be added to the array in the same order that they are arranged on the album, how would Sergiu access the name of the fifth track on the album?

- ○ album.track_name[4]
- ○ MusicRecord[5].track_name
- ○ album[4].track_name
- ○ album.track_name[5]
- ○ album[5].track_name

Quiz:
STEM SMART Computer Science Week 6

Dictionary: definition 2

Practice 1

In a dictionary, items are accessed by [   ], whereas items stored in an array are accessed by [   ]. A dictionary is sometimes called an [   ] array.

Arrays are useful when you want to [   ] over all the stored values because they are stored [   ] in main memory. Dictionaries are a better choice if you want to access discrete values because you retrieve them by [   ] rather than by [   ].

Items:

[value]   [iterate]   [ordered]   [key]   [contiguously]   [index]   [associative]   [randomly]

Quiz:

STEM SMART Computer Science Week 6

Dictionaries: compared to arrays

Challenge 1

What is the primary difference between an array and a dictionary in computer science?

○ Arrays can only store primitive data types, while dictionaries can store objects and structures.

○ Arrays use numeric indices whereas data in a dictionary is retrieved by key.

○ Arrays store data in key/value pairs, while dictionaries store data as a list of elements.

Raspberry Pi Foundation