1	1	**Mark is AO2 (analysis)** (The set of) integer (numbers) // $\mathbb{Z}$ // Z // the same set as the domain;	1
1	2	**Mark is AO2 (apply)** 4;	1
1	3	**2 marks AO1 (understanding) and 1 mark AO2 (analysis)** **2 AO1 (understanding) marks for describing how partial application works:** The function/add is applied to one of its arguments/4/6; **A.** one of the arguments is fixed/bound **A.** the new function takes one less argument // the new function takes one argument The output of this function application is a new function; **A.** a new function is created **1 AO2 (analysis) mark for working out what the output of the partial application would be in this instance:** The new function always adds the argument that the original function was applied to/4/6 to one argument; **A.** award this mark if it is clear what the new function does from its name eg "add4". **Note:** If a specific value (4 or 6) is used this must be the same value (4 or 6) for which the earlier mark point was awarded, if it was. Accept answers given by example, mapping the relevant part of the example to the mark scheme points. **Example of an answer by example** add6(x) **2 AO1 marks** for: • output of the partial application is a new function (eg add6) • the new function only has one argument(eg x) where add6(x) = 6 + x **1 AO2 mark** for explaining what new function does. **Max 2** for an answer by example if there is no description. *Answers can be a mix of marks for description and example.*	3

Qu	Pt	Marking guidance	Total marks
2	1	**All marks AO1 (understanding)** Immutable data structures // the state of a data structure cannot be changed (after creation); Statelessness // functions do not have side-effects // all functions are pure; Functions can be distributed to servers and executed on data sets then the results can be combined // map-reduce; Higher-order functions can compose the results of processing on multiple processors/cores // functions are first-class objects; The order of execution can be determined at run-time // the order of execution can be determined by the translator/compiler/interpreter (**A.** language) // the order of execution is not defined by the program code // programs are not a sequence of instructions that must be followed in a specific order; **NE.** Suitable for parallel processing **Max 2**	2

Qu	Pt	Marking guidance	Total marks
3	1	**All marks AO2 (analysis)**	3

Call number	Argument	Value returned
1	[4, 2, 5, 3]	52
2	[2, 5, 3]	24
3	[5, 3]	11
4	[3]	3
5	[]	0

1 mark: Argument column contains the following three values in order:
- [2, 5, 3]
- [5, 3]
- [3]

A. Destructuring argument to 2:[5, 3], 5:[3], 3:[]
R. Missing brackets

1 mark: Bottom row of table has **Argument** as [] and **Value returned** as 0
A. Argument column empty as **BOD**

1 mark: Value returned column contains 52, 24, 11, 3 in order
I. Working shown in column

Qu	Pt	Marking guidance	Total marks
3	2	**Mark is AO2 (analysis)** **A**; (The set of integers) **R.** if more than one lozenge shaded	1

Qu	Pt	Marking guidance	Total marks
4	1	**Mark is AO2 (analyse)** The set of real numbers // the set of all possible real-world quantities; **A.** real numbers **A.** numbers that represent any quantity along an infinite number line **A.** all numbers excluding imaginary/complex numbers **A.** rational and irrational numbers **NE.** $\mathbb{R}$ **TO.** real number stated but then another set described	**1**