User Defined Datatype

Papers Dock

COMPUTER SCIENCE 9618 PAPER 3

User Defined Datatype

Data types are like tools in a toolbox, each designed for a specific task in programming, dictating how data is stored and manipulated. Just as you wouldn't use a hammer to tighten a screw, each data type is suited to handle particular kinds of data (e.g., numbers, text) efficiently and appropriately.

What is meant by user defined datatype ?

- A data type constructed by a programmer and is not a primitive datatype (basic data types)
- A data type that references at least one other data type and is derived from one or more existing data type.
- The data types which are referenced can be primitive or user defined.
- To meet’s programmer requirement

Explain why user defined datatype are necessary ?

- To create a new data type from existing data types.
- To allow data types not available in a programming language to be constructed that meets programmer’s requirement

Datatypes

Non Composite	Composite
Single data type that does not refer to another data type	Data type that refers to other data types and are constructed from another datatype
Examples : Enumerated, Real, String, Char, Boolean, Integer	Examples : Record, List, Set, Array, Class, Queue, Linked List, Dictionary

Composite data types are like toolkits, combining multiple tools (e.g., lists, arrays, structs) to hold and manage collections of related data, whereas non-composite (primitive) data types are like single tools (e.g., int, float, char) designed to handle one specific type of data. Composite types bundle these individual tools together to solve more complex tasks, while non-composite types focus on simpler, specific tasks.

Explain what is meant by Composite Data Type ?

- A user defined data type that is a collection of data that can consist of multiple elements of different or the same data types.
- grouped under a single identifier

Explain what is meant by Non Composite Data Type ?

- It can be defined without referencing another data type.
- It can be primitive type available in a programming language, or a user-defined datatype.

Enumerated Datatypes

A user defined non composite data type with a list of all possible values that is ordered

Pseudocode To Declare Enumerated Datatype

```
TYPE name of Datatype = ( ____ , ____ , ____ , ____ )
```

Declare an enumerated data type with the identifier Months which can have all 12 months.

```
TYPE Months = ( January , February , March, April, ...... )
```

Declare a variable currentmonth of Datatype Months

```
DECLARE  currentmonth : Months
```

Assign August in currentmonth variable

```
currentmonth <--- August
```

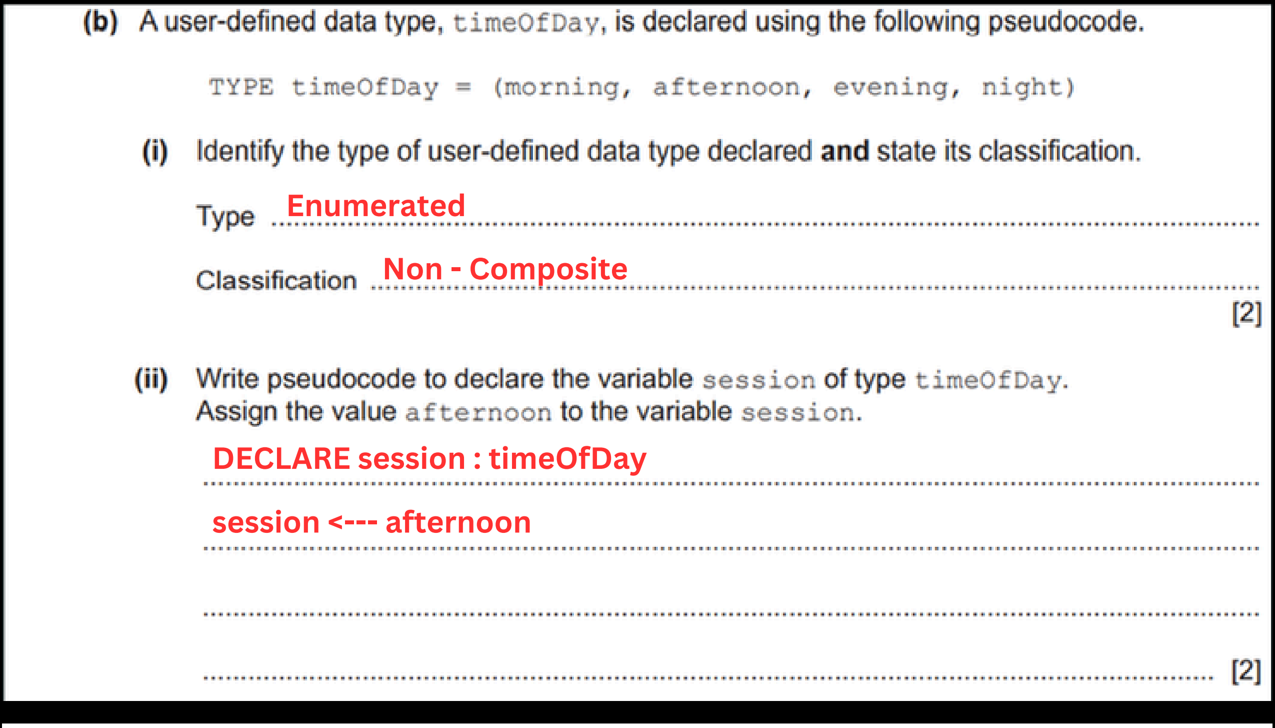
Declare a variable previousmonth of Datatype Month

```
DECLARE previousmonth : Months
```

Print the previous month

```
previousmonth <--- currentmonth - 1
OUTPUT previousmonth
```

Exam Style Question



(b) A user-defined data type, timeOfDay, is declared using the following pseudocode.

```
TYPE timeOfDay = (morning, afternoon, evening, night)
```

(i) Identify the type of user-defined data type declared **and** state its classification.

Type Enumerated

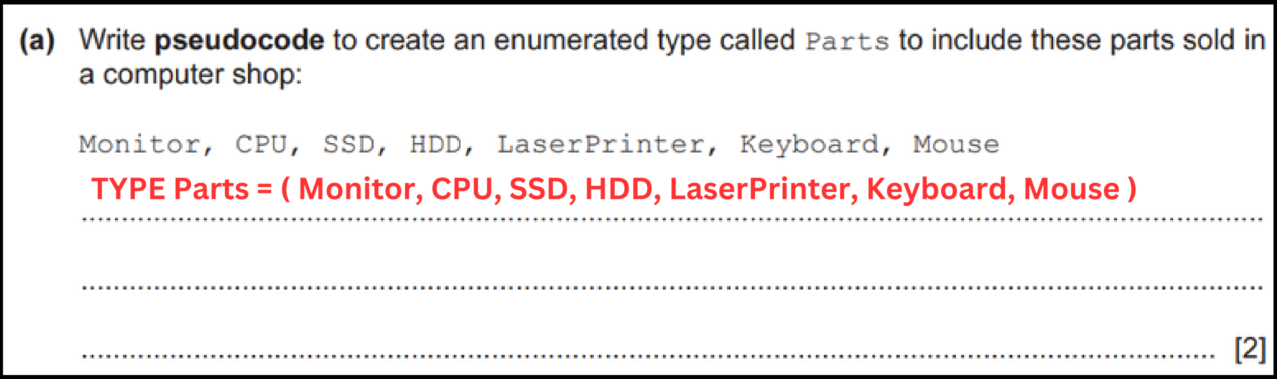
Classification Non - Composite

[2]

(ii) Write pseudocode to declare the variable session of type timeOfDay.
Assign the value afternoon to the variable session.

```
DECLARE session : timeOfDay
session <--- afternoon
```

[2]

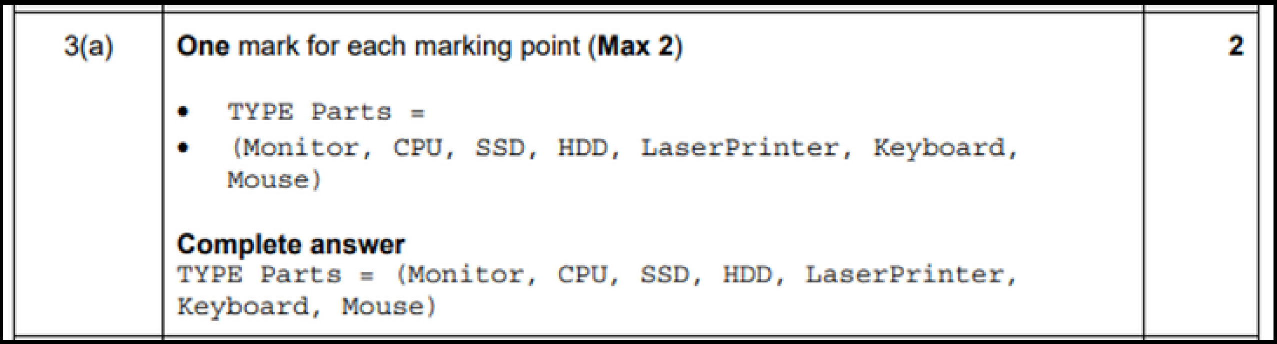


(a) Write **pseudocode** to create an enumerated type called Parts to include these parts sold in a computer shop:

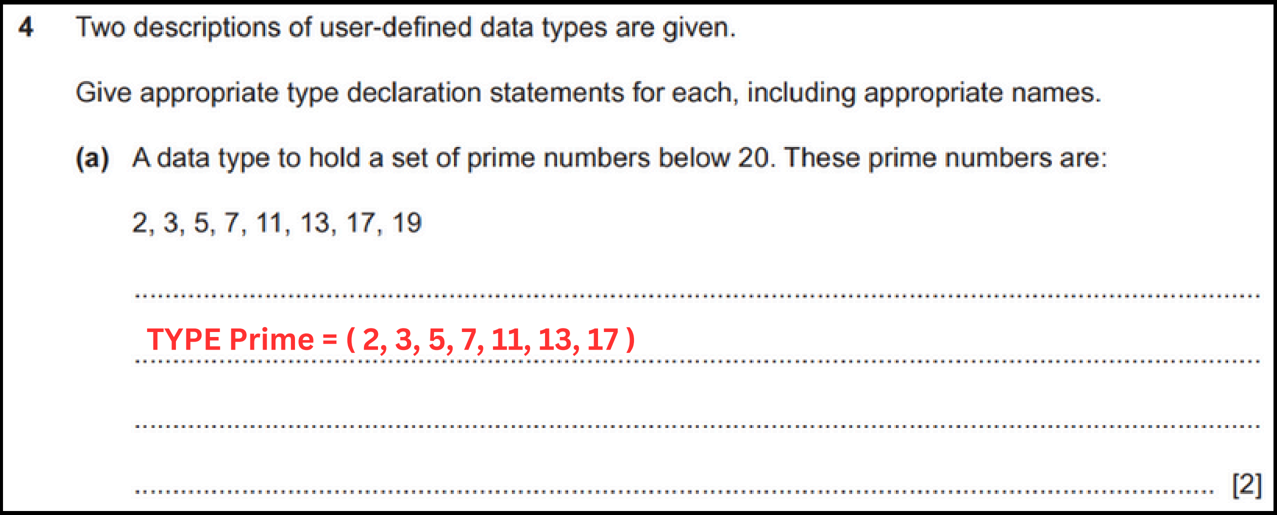
Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse

```
TYPE Parts = ( Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse )
```

[2]



3(a)	**One** mark for each marking point (**Max 2**) • TYPE Parts = • (Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse) **Complete answer** TYPE Parts = (Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse)	2



4 Two descriptions of user-defined data types are given.

Give appropriate type declaration statements for each, including appropriate names.

(a) A data type to hold a set of prime numbers below 20. These prime numbers are:

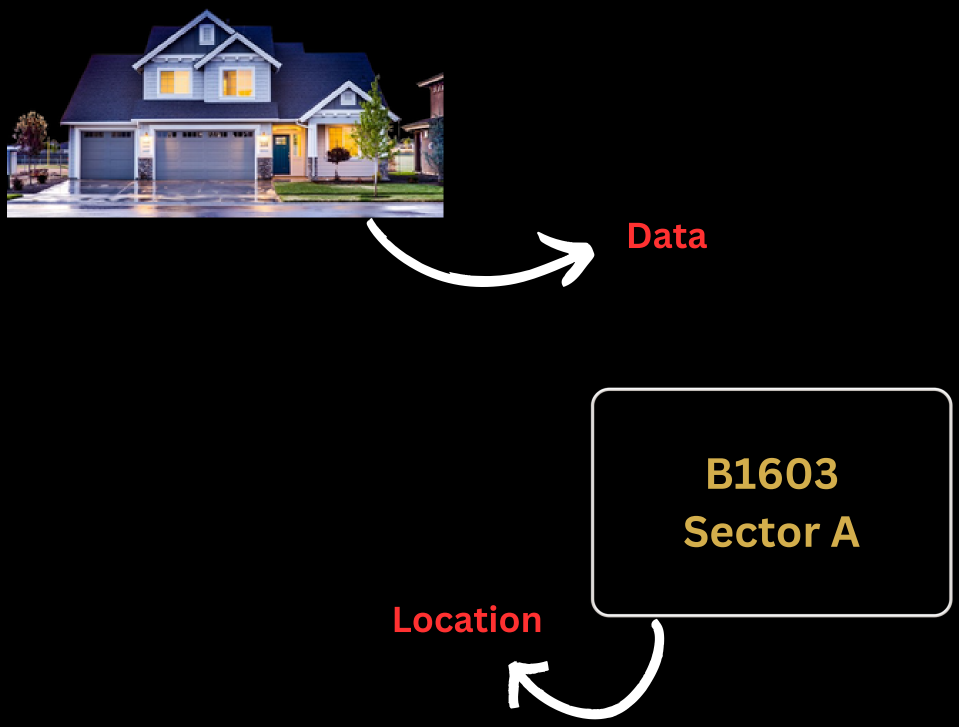
2, 3, 5, 7, 11, 13, 17, 19

```
TYPE Prime = ( 2, 3, 5, 7, 11, 13, 17 )
```

[2]

Pointer Datatypes

A user defined non composite datatype that stores memory locations only and indicates the type of data stored in the memory location.



Data

B1603
Sector A

Location

A pointer is like the address of a house. Just as the address tells you where the house is located, a pointer tells you where a variable's value is stored in memory. Instead of holding the house (data) itself, the pointer holds the address (memory location) where you can find the house (data).

Memory Location	Content
8216	
8217	“Ahmed”
8218	
8219	

we are going to make a variable to store the location 8217. So that variable would need a specific datatype known as pointer.

Base String

Pseudocode To Declare Pointer Datatype

^ : This symbol represents pointer

@ : This symbol represents the address is required not the data

```
TYPE name of Datatype = ^Base Datatype
```

Depends on the data stored so it could be String, Integer, Real, Boolean, Char

Question : We have Integer variables so create a data type with the name IntegerPointer

```
TYPE IntegerPointer = ^INTEGER
```

Declare a variable in which you store the address of integer values with the name myintegerpointer

```
DECLARE myintegerpointer : IntegerPointer
```

Question : You have a variable Number with value 10 in it stored on 5216 location. Store the address in myintegerpointer variable

```
myintegerpointer <---- @Number
```

(address of number not the value)

DEREFRENCING

You have the address and you want the value on that address

Change the data which is currently pointed by myintegerpointer to 100

```
myintegerpointer ^ <--- 100
```

(c) A pointer is a variable that stores the address of a variable of a particular type.

Consider the code on page 3, which uses the following identifiers:

Identifier	Data type	Description
`IntPointer`	`^INTEGER`	pointer to an integer
`IntVar`	`INTEGER`	an integer variable
`Temp1`	`INTEGER`	an integer variable
`Temp2`	`INTEGER`	an integer variable

Variable	Memory address	Contents
	...	
	8217	
`IntVar`	8216	`88`
	8215	
	8214	
	...	
	7307	
`IntPointer`	7306	`8216`
	7305	
	...	
	6717	
`Temp1`	6716	`88`
`Temp2`	6715	`57`
	6714	
	...	

Use the diagram to state the current values of the following expressions:

(i) `@Temp2` ....[1]

(ii) `IntPointer` ....[1]

(iii) `IntPointer^` ....[1]

(iv) `IntPointer^ = Temp2 + 6` ....[1]

(d) Write pseudocode statements that will achieve the following:

(i) Assign the value `22` to the variable `Temp2`.

....[1]

(ii) Place the address of `Temp1` in `IntPointer`.

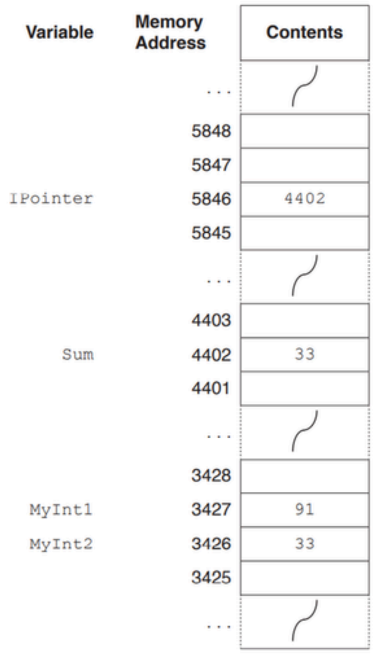
....[1]

(iii) Copy the value in `Temp2` into the memory location currently pointed at by `IntPointer`.

....[1]

```
Sum ← 91            // assigns the value 91 to the integer variable Sum
IPointer ← @Sum     // assigns to IPointer the address of the
                    // integer variable Sum
MyInt1 ← IPointer^  // assigns to variable MyInt1 the value at an
                    // address pointed at by IPointer
IPointer^ ← MyInt2  // assigns the value in the variable MyInt2 to
                    // the memory location pointed at by IPointer
```

The four assignment statements are executed. The diagram shows the memory contents after execution.



Variable	Memory Address	Contents
	...	
	5848	
	5847	
IPointer	5846	4402
	5845	
	...	
	4403	
Sum	4402	33
	4401	
	...	
	3428	
MyInt1	3427	91
MyInt2	3426	33
	3425	
	...	

Use the diagram to state the current values of the following expressions:

(i) `IPointer` ....[1]

(ii) `IPointer^` ....[1]

(iii) `@MyInt1` ....[1]

(iv) `IPointer^ = MyInt2` ....[1]

(d) Write pseudocode statements that will achieve the following:

(i) Place the address of `MyInt2` in `IPointer`.

....[1]

(ii) Assign the value `33` to the variable `MyInt1`.

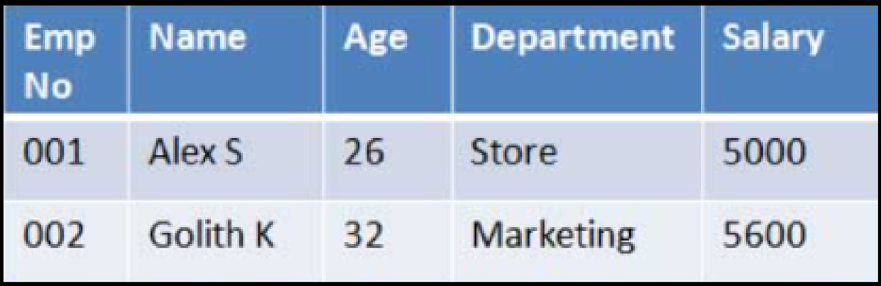
....[1]

(iii) Copy the value in `MyInt2` into the memory location currently pointed at by `IPointer`.

....[1]

Record Datatypes

It’s a composite data type and a group of multiple data types



Emp No	Name	Age	Department	Salary
001	Alex S	26	Store	5000
002	Golith K	32	Marketing	5600

If you want to store a complete record in a single variable name we use the concept of record datatype

Pseudocode To Declare Record Datatype

```
TYPE name of Datatype
        DECLARE Value1 : Datatype
        DECLARE Value2 : Datatype
ENDTYPE
```

Question : We need to store information of a book under a single identifier. Create a record datatype with the name Book. The book should hold info about ISBN number (Integer), Title (String), Genre (String).

Question : Declare a new record variable named MyBook

Question : The value of ISBN is 124657, Title is “Papersdock”, Genre is “Fiction”

Question : We need to store information of a book under a single identifier. Create a record datatype with the name Book. The book should hold info about ISBN number (Integer), Title (String), Genre (String).

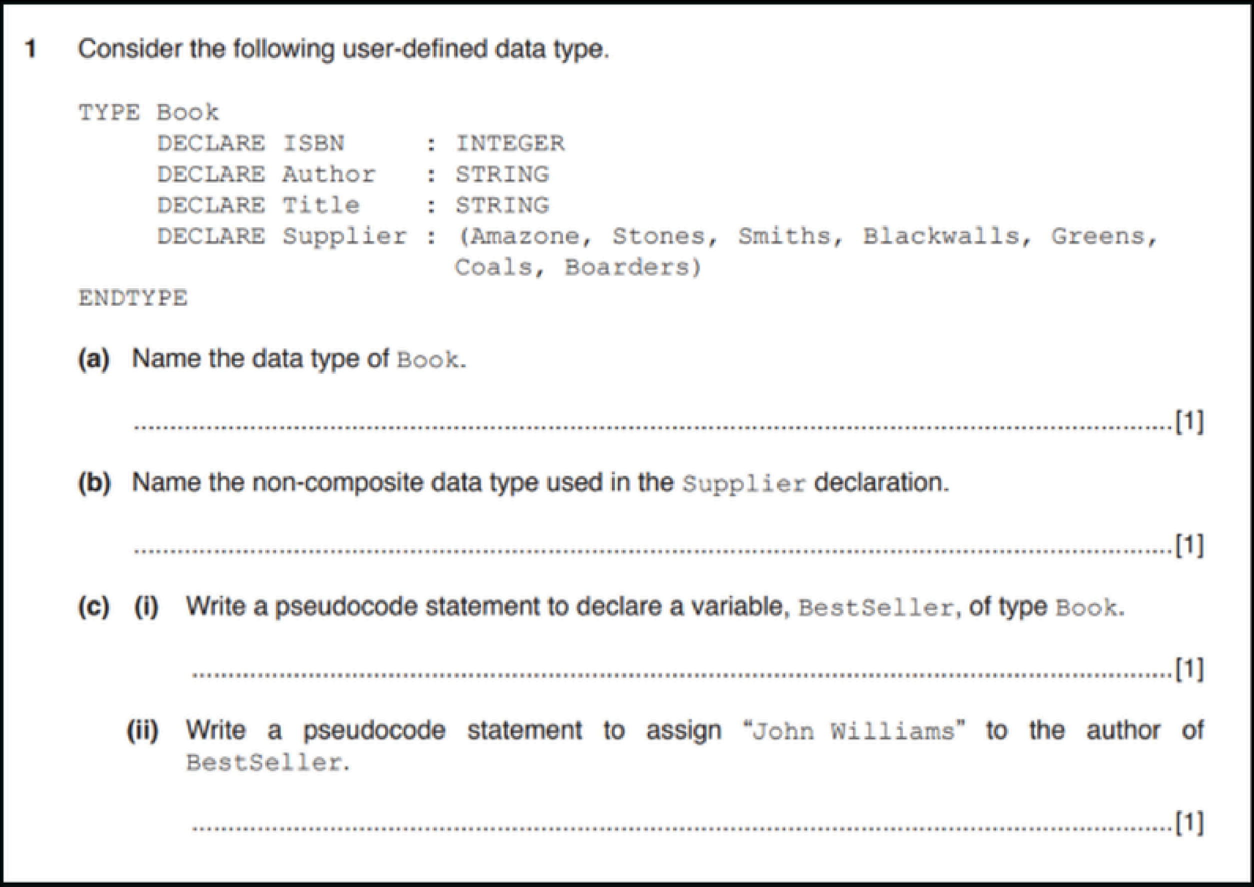
```
TYPE Book
    DECLARE ISBN : INTEGER
    DECLARE Title : STRING
    DECLARE Genre : STRING
ENDTYPE
```

Question : Declare a new record variable named MyBook

```
DECLARE MyBook : Book
```

Question : The value of ISBN is 124657, Title is “Papersdock”, Genre is “Fiction”

```
MyBook.ISBN <--- 124657
MyBook.Title <--- “Papersdock”
MyBook.Genre <--- “Fiction”
```



1 Consider the following user-defined data type.

```
TYPE Book
     DECLARE ISBN     : INTEGER
     DECLARE Author   : STRING
     DECLARE Title    : STRING
     DECLARE Supplier : (Amazone, Stones, Smiths, Blackwalls, Greens,
                         Coals, Boarders)
ENDTYPE
```

(a) Name the data type of `Book`.

....[1]

(b) Name the non-composite data type used in the `Supplier` declaration.

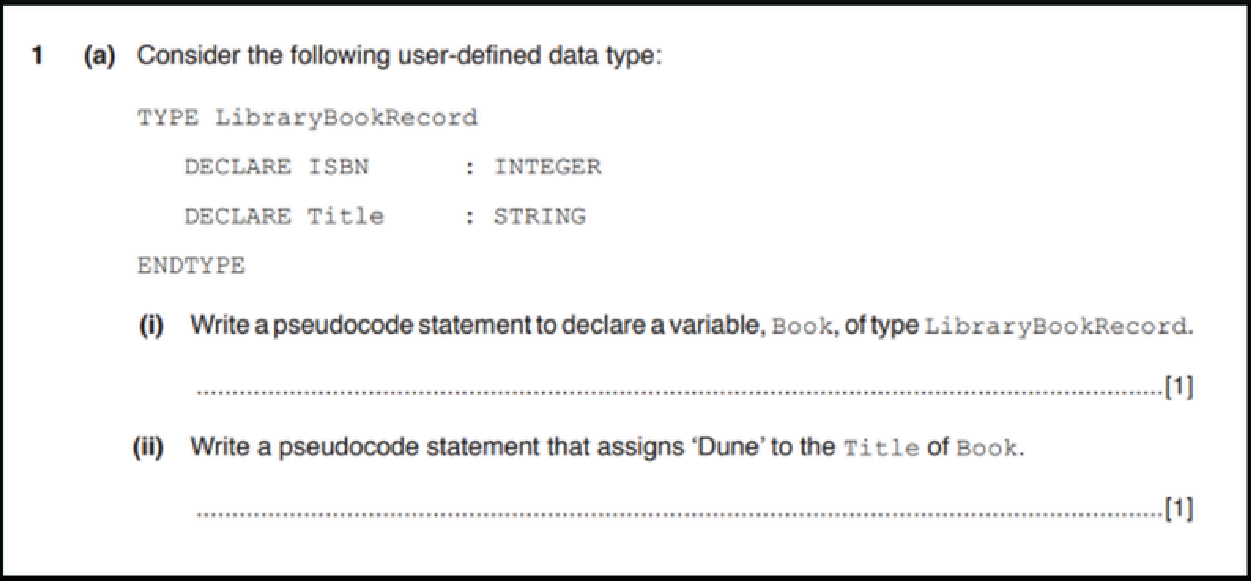
....[1]

(c) **(i)** Write a pseudocode statement to declare a variable, `BestSeller`, of type `Book`.

....[1]

(ii) Write a pseudocode statement to assign "`John Williams`" to the author of `BestSeller`.

....[1]



1 **(a)** Consider the following user-defined data type:

```
TYPE LibraryBookRecord
   DECLARE ISBN      : INTEGER
   DECLARE Title     : STRING
ENDTYPE
```

(i) Write a pseudocode statement to declare a variable, `Book`, of type `LibraryBookRecord`.

....[1]

(ii) Write a pseudocode statement that assigns 'Dune' to the `Title` of `Book`.

....[1]

Declaring A Range

Question : Declare a variable named Number which contains 0 till 99 numbers

```
DECLARE Number : 0 .. 9
```

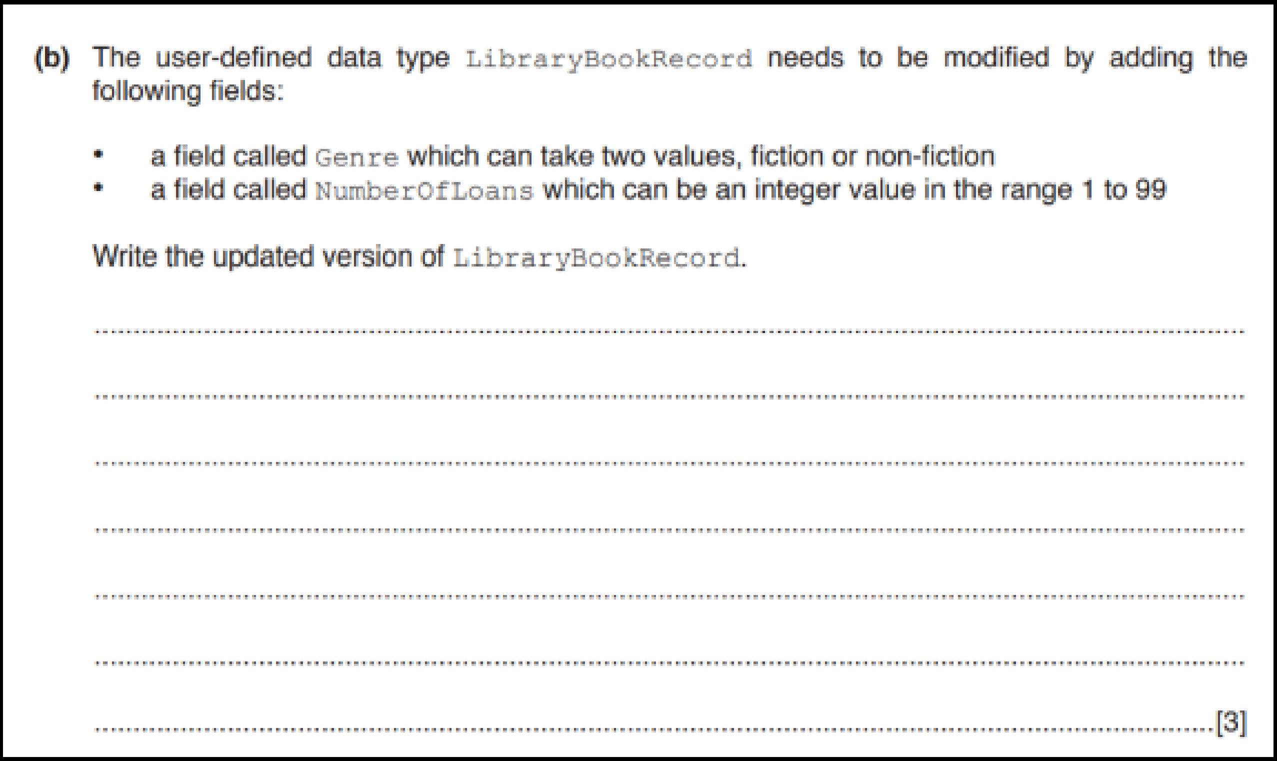
Specific Datatype In Array

If Array is declared with the data type String that means any string value can be stored in the array but if you want to make it specific values only then you are suppose to specify the values that could be assign to elements.

Question : Declare an Array that can only have “Taha” “Bano” “Pappan”

```
DECLARE Names : Array [ 1 : 3 ] OF ( “Taha”, “Bano” , “ Pappan” )
```

Note : If there are more than one thing to store than declare array and use Specific Datatype (with Quotation Marks) Average always means real values. If you have single value then you would need Enumerated (without Quotation Marks). Enumerated types are ideal for situations where you have a limited set of related values.



(b) The user-defined data type `LibraryBookRecord` needs to be modified by adding the following fields:

- a field called `Genre` which can take two values, fiction or non-fiction
- a field called `NumberOfLoans` which can be an integer value in the range 1 to 99

Write the updated version of `LibraryBookRecord`.

....

....

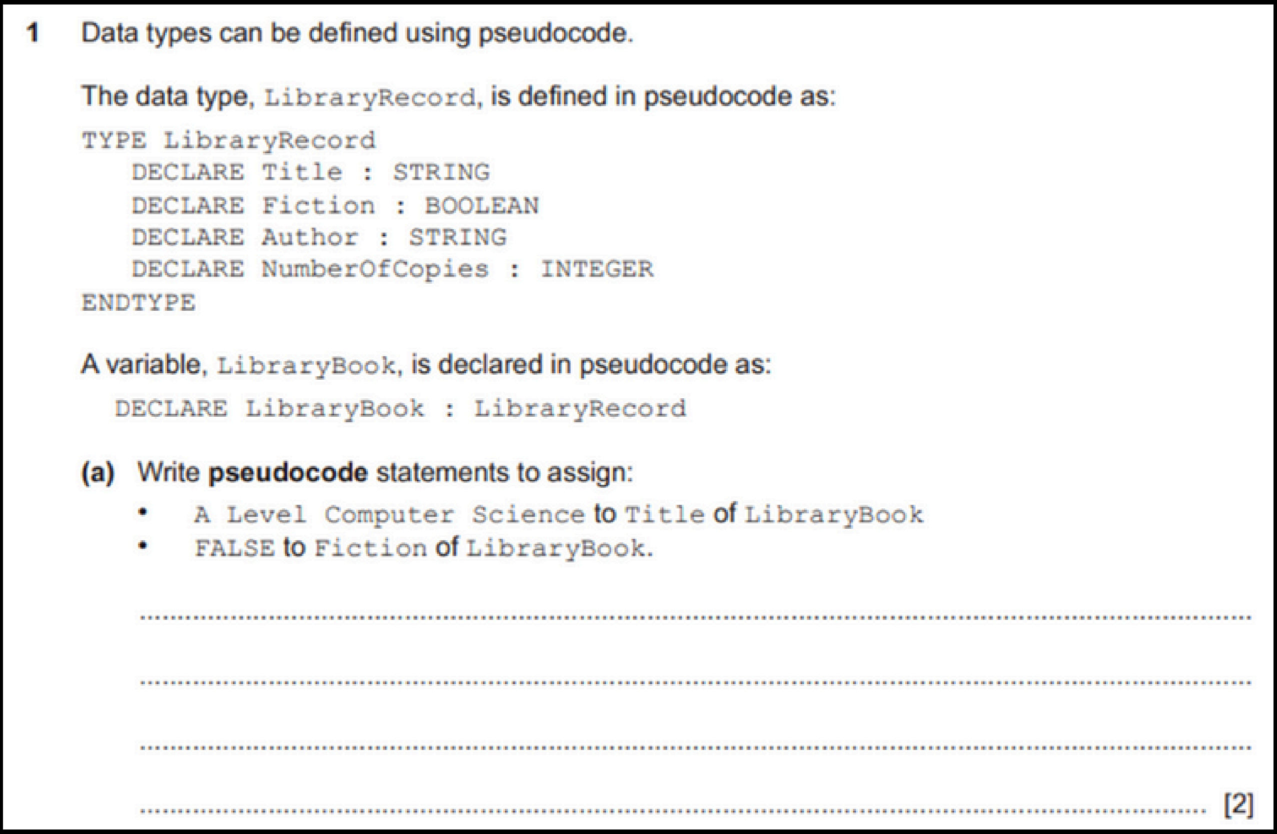
....

....

....

....

....[3]



1 Data types can be defined using pseudocode.

The data type, `LibraryRecord`, is defined in pseudocode as:

```
TYPE LibraryRecord
   DECLARE Title : STRING
   DECLARE Fiction : BOOLEAN
   DECLARE Author : STRING
   DECLARE NumberOfCopies : INTEGER
ENDTYPE
```

A variable, `LibraryBook`, is declared in pseudocode as:

```
  DECLARE LibraryBook : LibraryRecord
```

(a) Write **pseudocode** statements to assign:

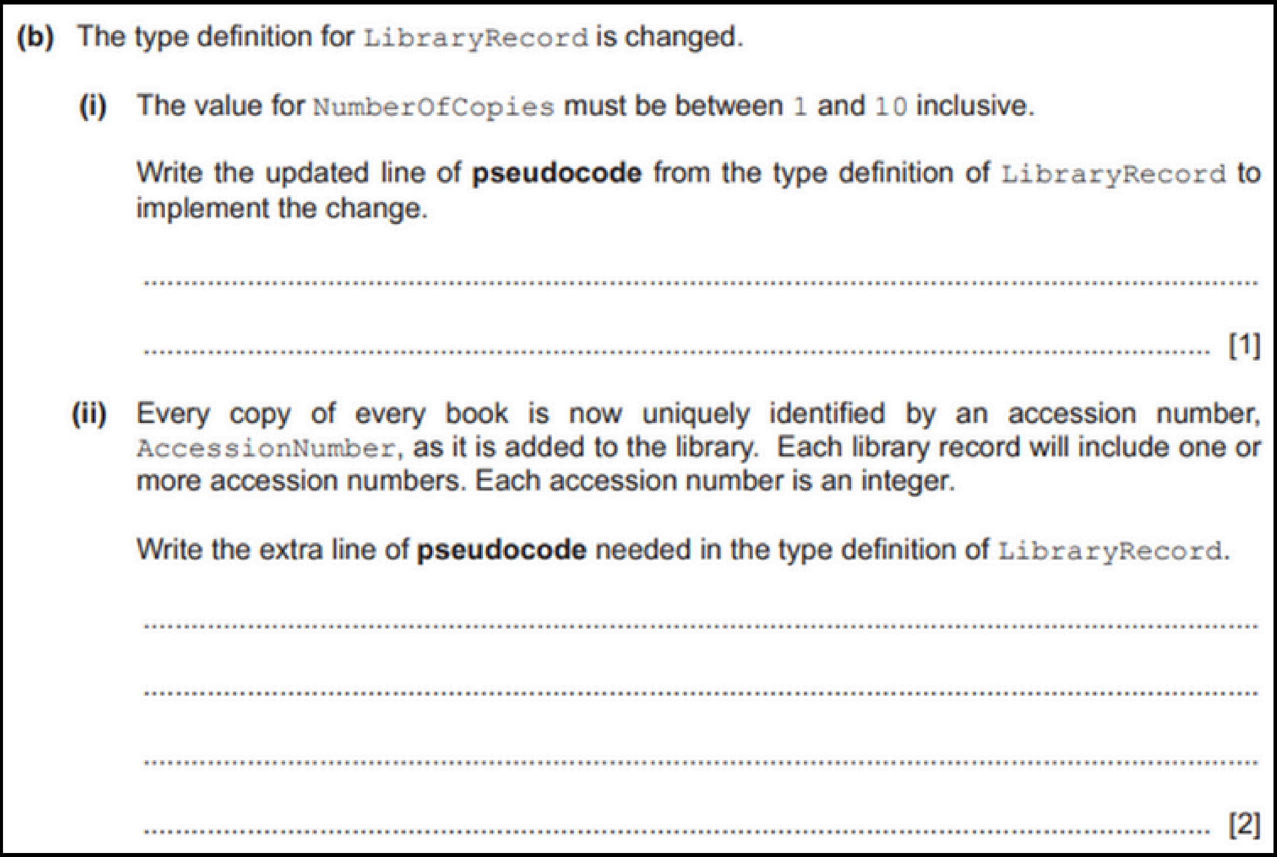
- `A Level Computer Science` to `Title` of `LibraryBook`
- `FALSE` to `Fiction` of `LibraryBook`.

....

....

....

.... [2]



(b) The type definition for `LibraryRecord` is changed.

(i) The value for `NumberOfCopies` must be between `1` and `10` inclusive.

Write the updated line of **pseudocode** from the type definition of `LibraryRecord` to implement the change.

....

.... [1]

(ii) Every copy of every book is now uniquely identified by an accession number, `AccessionNumber`, as it is added to the library. Each library record will include one or more accession numbers. Each accession number is an integer.

Write the extra line of **pseudocode** needed in the type definition of `LibraryRecord`.

....

....

....

.... [2]