Ch13

Data Types and Structures

CHAPTER ASSIGNMENT · 25 MARKS

Name _____ Class _____ No. _____ Date _____

Answer every question in the space given. Show your working. You may use the chapter page to help you.

Questions from *Computer Science for Cambridge International AS & A Level Coursebook*, 2nd edition, Sylvia Langfield and Dave Duddell (Cambridge University Press), for classroom use.

Tasks and questions from the chapter

13.01 Look at the identifier tables in Chapter 12 (Tables 12.06 and 12.09 to 12.12). Give the data type that is appropriate for each variable listed.

13.02 Write the declaration of a record type to store the details of a book: Title, Year of publication, Price, ISBN.

Write the statements required to assign the values "Computer Science", 2019, £44.95, "9781108733755" to the fields respectively.

13.03 Define two arrays, one for your friends' names and one for their ages as shown in Figure 13.02.

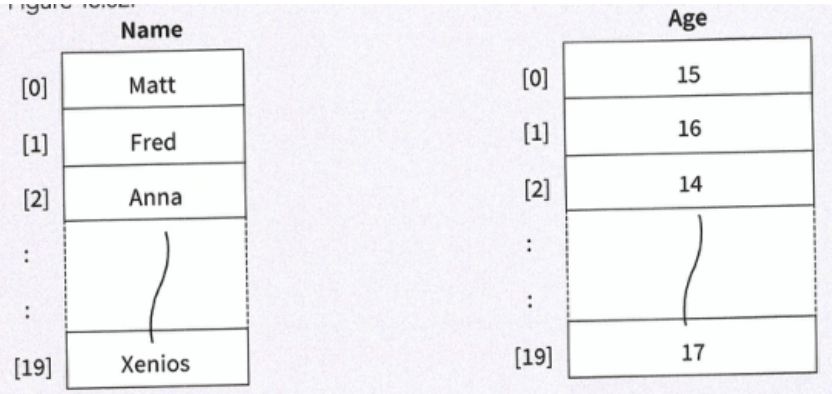


Figure 13.02 Arrays for names and ages

13.04 Use the algorithm in Worked Example 13.02 as a design pattern. Write an algorithm using the arrays from Task 13.03 to search for a friend's name and output their age.

DP **Discussion Point:** What happens if the array elements are already in the correct order?

13.05 Rewrite the algorithm in Worked Example 13.03 to sort the array elements into descending order.

13.06

(a) Declare a 2D array to store the board data for the game Noughts and Crosses. The empty squares of the board are to be represented by a space. Player A's counters are to be represented by "O". Player B's counters are to be represented by "X".

(b) Initialise the array to start with each square being empty.

(c) Write a statement to represent player A placing their counter in the top left square.

(d) Write a statement to represent player B placing their counter in the middle square.

13.07

(a) Declare an array of `BookType` (see Task 13.02) for 200 books.

(b) Set the first book's details to the values given in Task 13.02.

13.08

(a) Write pseudocode to save the array data from Task 13.06 to a text file.

(b) Write pseudocode to read the values stored in the text file back into the board array.

13.09

(a) Draw a diagram to show the contents of the stack shown in Figure 13.07 after "E" has been pushed onto the stack.

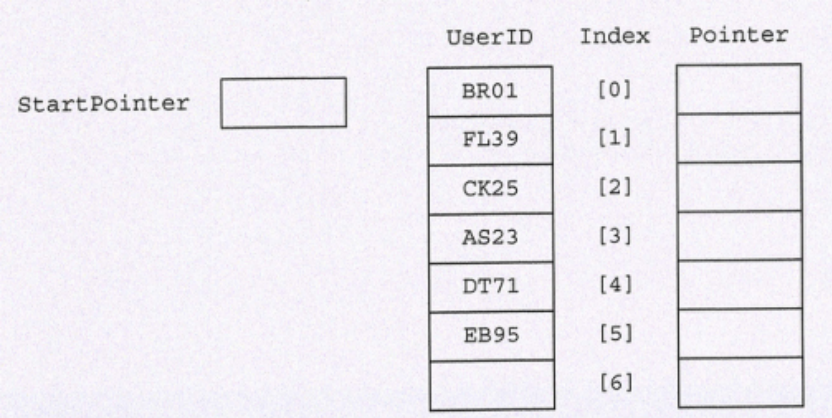
(b) Draw a diagram to show the contents of the stack shown in Figure 13.07 after one item has been popped off the stack.

13.10

(a) Draw a diagram to show the contents of the queue after "F" has joined the non-empty queue shown in Figure 13.08.

(b) Draw a diagram to show the contents of the queue after one item has left the non-empty queue shown in Figure 13.08.

13.11 A linked list is to be set up using the values in the `UserID` array shown below.



Without moving any of the contents of the `UserID` array, insert the pointer values so that the linked list is in alphabetical order.

RP **Reflection Point:** What is the difference between standard data types, ADTs and user-defined data types?

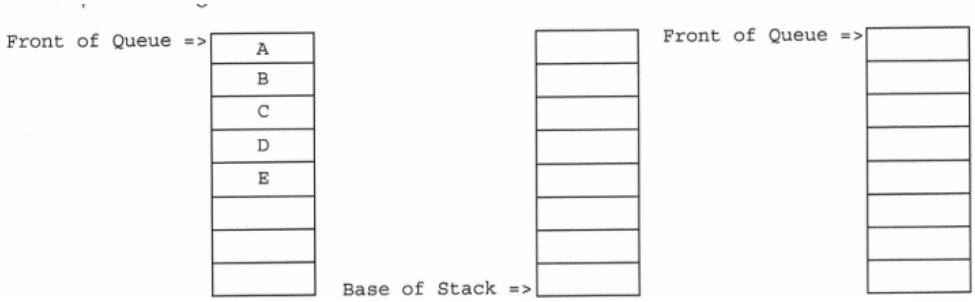
Exam-style questions

1 Complete the following variable identifier table: [3]

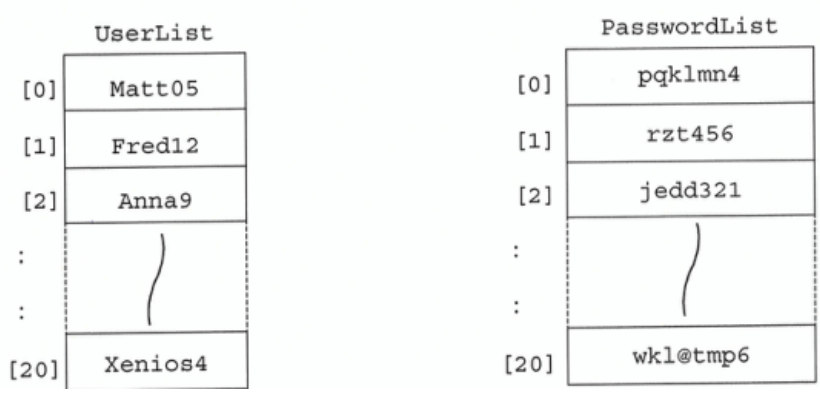
Variable	Example value	Data type
ColourCode	"034AB45"	
ProductionDate	2018/03/31	
Weight	67.45	
NumberInStock	98	
SizeCode	'X'	
Completed	FALSE	

2 A stack and a queue are used to reverse the order of a set of values.

Complete the diagram: [2]



3 Alicia uses two 1D arrays, `UserList` and `PasswordList`. For twenty users, she stores each user ID in `UserList` and the corresponding password in `PasswordList`. For example, the person with user ID `Fred12` has password `rzt456`.



Alicia wants to write an algorithm to check whether a user ID and password, entered by a user, are correct. She designs the algorithm to search `UserList` for the user ID. If the user ID is found, the password stored in `PasswordList` is to be compared to the entered password. If the passwords match, the login is successful. In all other cases, login is unsuccessful.

(a) Complete the identifier table. [8]

Identifier	Data type	Explanation
UserList		1D array to store user IDs
		1D array to store passwords
MaxIndex		Upper bound of the array

Identifier	Data type	Explanation
MyUserID		User ID entered to login
MyPassword		
UserIdFound		FALSE if user ID not found in UserList; TRUE if .....
LoginOK		FALSE if .....; TRUE if .....
Index		Pointer to current array element

(b) Complete the pseudocode for Alicia's algorithm: [10]

```
MaxIndex <- 20
INPUT MyUserID
INPUT MyPassword
UserIdFound <- FALSE
LoginOK <- ..............
Index <- -1
REPEAT
    INDEX <- ..............
    IF UserList[..............] = ..............
      THEN
        UserIdFound <- TRUE
    ENDIF
UNTIL ..............  OR ..............
IF UserIdFound = TRUE
  THEN
    IF PasswordList[..............] = ..............
      THEN
        LoginOK <- TRUE
    ..............
ENDIF
IF ..............
  THEN
    OUTPUT "Login successful"
  ELSE
    OUTPUT "User ID and/or password incorrect"
ENDIF
```

(c) Instead of using two 1D arrays, Alicia could have used an array of records.

(i) Write pseudocode to declare the record structure `UserRecord`. [2]

(ii) Write pseudocode to declare the `User` array. [2]

End of assignment · Check your answers against the chapter at gurujazzy.com/subjects/computer-science/as-a-level/data-types-structures/