|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **1** | **2** | **3** | **4** | **Total** |
| *maximum* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Arrays and Linked Structures**

Reading Assignment: Thoroughly read Chapter 4 in the course textbook.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part 1 Glossary Terms**

Define, in detail, each of these glossary terms from the realm of computer programming logic and design and computer topics, in general. If applicable, use examples to support your definitions. Consult your notes or course textbook(s) as references or by visiting Web sites such as: [**http://www.ask.com**](http://www.ask.com),[**http://www.bing.com**](http://www.bing.com), [**http://www.webopedia.com**](http://www.webopedia.com)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(a) Doubly Linked Structures**

|  |
| --- |
| A doubly linked structure has two-way nodes with a ‘previous’ link as well as a ‘next’ link, and also have a “tail” node that always points to the last node in the structure. |

**(b) Logical Size ( of an array )**

|  |
| --- |
| The logical size of an array is the number of items in an array that should be available to the application. |

**(c) Physical Size ( of an array )**

|  |
| --- |
| The physical size of an array is the total number of array cells that is used to specify its capacity when the array is created. |

**(d) Ragged Grid**

|  |
| --- |
| A ragged grid is a two dimensional, non-rectangular grid that may have empty or null items in its cells. It has a fixed number of rows but the number of columns can vary. |

**(e) Singly Linked Structures**

|  |
| --- |
| A singly linked structure is a linear sequence of items in which each node (which is a basic unit of representation for items in a linked structure) contains a data item as well as a pointer (or, reference) to another node. Nodes in linked structures contain references (pointers) to an area of non-contiguous memory called an “object heap”, and links within nodes can point to any location in this area of memory. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part 2 True / False Exercises**

For each of these exercises, enter True or False in the spaces provided.

**TRUE** **(1)** To access a two - dimensional array, you use two subscripts.

**TRUE**  **(2)** In a doubly linked structure, the first and last item have an empty link.  
 **this is true in that the previous link is empty in the first node, and the next link is empty in the last node**

**TRUE (3)** The insertion and removal of the first node of a singly linked structure require that the head pointer be reset.

**TRUE** **(4)** A circular linked structure contains a link from the last node back to the first node in the structure.

**Tru-ish, in that the last node contain a link back to a dummy header, which is in fact the first node in the structure…**

**FALSE** **(5)** The run - time complexities of the operations on a doubly linked structure are typically double compared to the corresponding operations on the singly linked structure.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part 3 Multiple Choice Exercises**

Select the correct response or responses.

**(1)** Which of the following best describes an array?

(a) a collection of data points that represent an object

(b) a list of values that are indexes to a database

(c) a numeric value that points to a position in RAM where data can be found

**(d) a sequence of items that can be accessed at given index positions**

**(2)** In the Array class defined in the textbook, how do you instantiate an array object that can hold 10 values?

(a) myArray(10) = Array (b) Array myArray, 10

**(c) myArray = Array(10)** (d) Array(10) myArray

**(3)** What method does Python's list type use to increase the size of the underlying array?

(a) size **(b) append**

(c) increase (d) augment

**Append doesn’t increase the size of the underlying array, rather, the Python List type has dynamic arrays baked in, so that when the load factor of an array reaches a certain threshold, a new array is created, and all the items from the old array are transferred to the new array and the old array is deleted (marked for pickup by Python’s garbage collector)**

**(4)** The process for resizing an array named **myArray** is shown below. What is the missing code?

**if logicalSize == len(myArray) :**

**temp = Array(len(myArray) + 1)**

**for i in range(logicalSize) :**

**# <missing code>**

**a = temp**

(a) myArray[temp] = myArray[i]

**(b) temp [i] = myArray[i]**

(c) myArray[i] = temp[i]

(d) temp = myArray(len(myArray))

**(5)** Which of the following statements accesses the second column in the third row of a two - dimensional array?

**(a) twoDim[2][1]**  (b) twoDim[4][3]

(c) twoDim[1][2] (d) twoDim[2][3]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part 4 Programming Exercises**

**(1)** **( The array Class Library )**

**Modify the above statements whereby instead of numbers, an array of characters is declared…**

*import* array *as* arr

characters = arr.array("u", ["a", "b", "c", "d", "e", "f", "g"])

*# change the first element*

characters[0] = "f"

print([char *for* char *in* characters])

*# output: ['f', 'b', 'c', 'd', 'e', 'f', 'g']*

*# change the 3rd to the 5th element*

characters[2: 5] = arr.array("u", ["h", "i", "j"])

print([char *for* char *in* characters])

*# Output: ['f', 'b', 'h', 'i', 'j', 'f', 'g']*

print(sum(characters))

**(2)** **( The array Class Library )**

*# Modify the program segment below such that the characters in your First Name and Last Name given to the arrays first and second . Then display your full name, in concatenated form.*

*import* array *as* myArr

firstName = myArr.array("u", ["h", "a", "y", "e", "s"])

lastName = myArr.array("u", ["c", "r", "o", "w", "l", "e", "y"])

name = myArr.array("u")

name = firstName + lastName

print(name)

*# output: array('u', 'hayescrowley')*

print([char *for* char *in* name])

*# output:*

*# ['h', 'a', 'y', 'e', 's', 'c', 'r', 'o', 'w', 'l', 'e', 'y']*

**(3) ( Count the Occurrences )**

*# Supplement the statements below by requesting a value from the user, who wishes to determine the number of occurrences of a specific number that may be an element of the array.*

*import* array *as* myArr

number = myArr.array("b", [2, 3, 5, 4, 3, 3, 3])

*try*:

countNumber = int(

input("What number would you like to count? Please enter an integer: "))

*except* ValueError:

print("please enter an integer!!")

print(number.count(countNumber))

*# output: entering 3 as an input prints 4, as 3 occurs 4 times in the array*

**(4) ( Multi - Dimensional Arrays )**

*# Determine the missing line of this program segment. The purpose of the omitted statement is to sum, i.e. accumulate, all the elements of the two - dimensional array.*

grid = [[1, 2, 3, 5, 7, 10], [6, 2, 4, 5, 9, 11]]

sum = 0

*for* row *in* range(len(grid)):

*for* column *in* range(len(grid[0])):

sum += grid[row][column]

print(sum)

*# output: 65*

**(5) ( Cryptography - The Rail Fence Cipher )**

*import* array *as* arr

msg = ["H", "e", "l", "l", "o", "S", "t", "u", "d", "e", "n", "t"]

str1 = []

str2 = []

print("Rail Fence - Encryption\n\n")

print("Plain Text: HelloStudent\n\n")

*for* i *in* range(len(msg)):

*if* (i % 2 == 0):

str1.append(msg[i])

*else*:

str2.append(msg[i])

print("Cipher Text:", "".join(str1 + str2))

*# output*

*# Rail Fence - Encryption*

*# Plain Text: HelloStudent*

*# Cipher Text: HlotdnelSuet*