

TCB2
(33)



THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE

EXAMINATIONS OF December 2018

Code and Name of Course: COMP1602 – Computer Programming II

Date and Time: Monday 10th December 2018 1pm

Duration: 2 Hours

INSTRUCTIONS TO CANDIDATES: This paper has 4 pages and 3 questions

Answer ALL Questions



- 1) a) Write a fragment of code to generate (i.e., not hard code) an alphabet chart as follows:

```

A a   B b   C c   D d   E e
F f   G g   H h   I i   J j
K k   L l   M m   N n   O o
P p   Q q   R r   S s   T t
U u   V v   W w   X x   Y y
Z z

```

[5 marks]

- b) Write a fragment of code (with declarations) that reads a pair of words from the keyboard and determines how many letters in corresponding positions of each word are the same. The words consist only of the letters of the English alphabet (uppercase and lowercase).

For example, if the words are *coat* and *CATTLE*, the following output should be generated:

```

c is a matching letter in position 0
t is a matching letter in position 3

```

There are 2 matching letter/s.

[8 marks]

- c) A *nym* word is a word which **ends** with the letters *nym*. For example, *antonym* and *homonym* are *nym* words. Write a function, *isNymWord*, with the following prototype:

```
bool isNymWord (char s[])
```

which accepts a C-string *s* and returns *true* if *s* is a *nym* word and *false*, otherwise.

Assume that *s* contains only lowercase letters. You may use the built-in *strlen* function.

[5 marks]

- d) A certain 6-sided die is weighted such that 2's and 4's come up twice as likely as any other number. Write a program to simulate 1000 throws of the die and output how many times each number occurs.

[8 marks]

Total marks: 26



2) a) A sorted integer array contains the following values:

0	1	2	3	4	5	6	7	8	9	10
22	25	33	46	50	55	69	70	74	80	86

- i) A *binary search* is used to search for the value 69 in the array. How many comparisons will be made before it is found? Show your working. [4 marks]
 - ii) Suppose that a linear search is made. How many *comparisons* will be made to find 69? What statement can you make about the relative efficiency of the binary search algorithm compared to the linear search algorithm on a sorted array? [2 marks]
- b) A *mysterySort* function is called with the array *A* and the array *C*, both of size *m*. The code for the *mysterySort* function is given below:

```
void mysterySort(int A[], int C[], int m){
    int lo = 0;
    int hi = m - 1;
    int k = 0;
    while (lo <= hi) {
        if (A[lo] > A[hi]) {
            C[k] = A[lo];
            lo = lo + 1;
        }
        else {
            C[k] = A[hi];
            hi = hi - 1;
        }
        k = k + 1;
    }
}
```

The array *A* is shown below.

<i>A</i>	152	87	24	8	12	17	20	78	112
----------	-----	----	----	---	----	----	----	----	-----

For each of the **first four (4)** passes of *mysterySort* draw:

- i) the array *A*, indicating the value of *lo* and *hi*, and
- ii) the contents of the array *C*. [8 marks]

Total marks: 14



- 3 a) Figure 1 below shows a two-dimensional array of integers, *m*:

0	1	2	3	4	5
1	2	3	4	5	6
2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	10

Figure 1

- i) Write code to declare the array *m*. [1 mark]
 - ii) The array shown in Figure 1 has already been loaded with values. Write a segment of code that can be used to assign the values shown in Figure 1 to the array. You **must** use nested '*for*' loops in your answer. [4 marks]
 - iii) Write a segment of code to interchange the values in column 0 with the values in column 5. [3 marks]
 - iv) Write a segment of code to *randomly* generate a location in the array and assign the value of -1 to that location in the array. [2 marks]
- b) A 2D array is called *top heavy* if the top half of the array contains *more non-zero* elements than the bottom half of the array. The 2D array must have an even number of rows. For example, the following 2D array is top heavy:

0	1	2
9	1	0
8	0	2
0	0	3

↑ top

bottom ↓

Write a function, *topHeavy*, which accepts a 2D array *m* with *numRows* rows and *numCols* columns as parameters (where the maximum number of rows and columns is 100), and returns *true* if *m* is top heavy. The function must return *false* if *m* is not top heavy or *numRows* is odd. [5 marks]

Total marks: 15

End of Examination
(Total marks = 55)