**CS278 “C” Test 3 Spring 2009 Burris**

Answer any four questions. Number your questions on the answer sheets in ascending numeric order from one through nine inclusive. Clearly write “delete” on the answer sheets by the five questions you do not wish graded. Leave at least a one-inch margin at the top of every page. Start each question you answer on a new page. You may however delete more than one consecutive question on the same page. Do not write on the back of pages. Staple your answer sheets (in ascending numeric order) on top of the test in the upper left hand corner. Write your name in the upper right hand corner of the answer sheets (first page). Turn the stapled bundle over and write your name in the upper right hand corner on the back of the test. All answers must be placed on the answer sheets. No credit will be awarded for work appearing on the test. A five point Road Map Fee (RMF) will be deducted for each instance of failure to follow instructions. You will not receive credit for material I can not read (illegible) or that is obstructed from my view, e.g., by a staple.

Warning: There are no “short” answers on this test. Tell me everything germane to the topic. Your performance is being compared to all other members of the class. . If you cannot write a page of relevant information, you may wish to consider another question.

1) What specific function does the following code accomplish? How long does it require to execute in terms of numInt? Express your answer using Big “O” notation for the performance as N 🡪 ∞. You must support your answer analytically to receive credit. How do keys with the same value on function entry appear on function exit?

**void Mystery( int intArray[ ], int numInt)**

**{**

**for(int j = 0; j < numInt - 1; j++ ){**

**for(int k = j + 1; k < numInt; k++ ){**

**if( \*(intArray + j) >= \*(intArray + k ) ){** .

**int temp = \*(intArray + j );**

**\*(intArray + j ) = \*(intArray + k);**

**\*(intArray + k ) = temp;**

**}**

**}**

**}**

**}**

2) Discuss the difference between call-by-value and call-by-reference in “C.” Include any limitations on data types that may be passed by value.

3) “C” allows two types of functions. Explain each type in detail and how they differ. Your discussion should include logical/physical restrictions on the use of parameters sent to the function and results returned.

4) The function “deleteItem” accepts an array of grades, the number of grades currently in the array, and the location the new grade is to be deleted. Write the function code to accomplish the deletion operation. Deletion must be accomplished by moving all grades below the deletion point up one. Do not destroy any existing grades. Prior to returning to the main program, the number of grades in the array should be decremented by 1 to reflect the array contents is now one smaller.

void deleteItem( int Grades[ ], int \*numGrades, int deletePoint);

As an example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Prior to deletion: |  |  |  | After insert: |  |
|  | Grades |  |  |  | Grades |  |
| 0 | 78 |  |  | 0 | 78 |  |
| 1 | 82 |  |  | 1 | 85 |  |
| 2 | 45 |  |  | 2 | 92 |  |
| 3 | 92 |  |  | 3 | 85 |  |
| 4 | 85 |  |  | 4 |  |  |
| 5 |  |  |  | 5 |  |  |
| 6 |  |  |  | 6 |  |  |
|  |  |  |  |  |  |  |
|  | numGrades = 5 |  |  |  | If deletePoint = 2, after deletion numGrades = 4 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

5) Given the following declarations, write a section of code to search the table for the desired item. If the item is found, print the price. If the item is not found, print an appropriate message. There are currently 356 items in inventory (locations 0 through 355). The desired item may appear in at most one bin. For full credit, use search method 3. You will receive most of the credit for search method 2, and half credit for search method 3.

struct bin {

char\* item;

double price;

};

struct bin table[ 500 ];

int numItems = 356

char \*desiredItem

6) Explain the difference between search methods one, two, and three. You must clearly delineate the circumstances under which each is better and why.

7) Explain the function accomplished by the following function.

**void mystry2(char str1[ ], char str2[ ]) {**

**int k = 0;**

**while ( str1[k] = str2[k] ) k++;**

**}**

8) Explain in detail the concept of an abstract data type (ADT). How are ADT’s related to templates? Rewrite the above question as a template accepting any intrinsic data type (char, int, long, float, double, byte, short, etceteras).

9) What is printed by the following section of code. Appropriate values are indicated by the comments.

#include <iostream>

using namespace std;

**int main( ){**

**int m, k,**

**\*pt1,**

**\*\*pt2;**

**\*\*\*pt3;**

**k = 1234;**

**m = 56;**

**pt1= &k; // pt1 = 0x0012FF26**

**pt2 = &pt1 // pt2 = 0x0012FF50**

**pt3 = &pt2; // pt3 = 0x0012FF47**

**\*\*pt2 = m;**

**\*pt1 = m + 6;**

**k = k + m;**

**cout << \*\*\*pt3 << endl;**

**}**