Final Exam

Introduction to Computers and Programming

Summer 2000

Name ____________________________

Rules

• NO GROUP EFFORTS

• Closed books, closed notes, no calculators

• Follow directions carefully

• Please print (not a rule, just a suggestion)

Note: Each problem will be worth between 20 and 30 points.

Problem 1 _____ / ______
Problem 2 _____ / ______
Problem 3 _____ / ______
Problem 4 _____ / ______
Total _____ / 100
1. **Define a function** called `locateFirstVowel` with the following function prototype:

   ```
   int locateFirstVowel(char str[]);
   ```

   The function takes one parameter, `str` (a character string), and returns the **index** of the **first** character in the string that is a lower case vowel (’a’, ’e’, ’i’, ’o’, or ’u’); if the string `str` contains no lower case vowels, the function must return the integer -1.

   For example:
   ```
   locateFirstVowel("abc") would return 0
   locateFirstVowel("tree") would return 2
   locateFirstVowel("xyz") would return -1
   ```

   Note: this function basically performs a linear search on the character array `str`.
2. There are many algorithms for sorting an array. Below is a C function that implements a sorting algorithm called insertion sort. Like bubble sort and selection sort, insertion sort makes a number of passes through the array to be sorted.

```c
void insertionSort(int a[], int size) {
    int pass, i, j, hold;

    for (pass = 1; pass < size; pass = pass + 1) {
        // determine where a[pass] goes
        i = 0;
        while (a[i] < a[pass])
            i = i + 1;

        hold = a[pass]; // save a[pass]

        // make room for a[pass] by moving elements
        // to the end of the array
        for (j = pass; j > i; j = j - 1)
            a[j] = a[j - 1];

        a[i] = hold; // insert a[pass]
    } // end outer for loop
} // end function insertionSort
```

Assume that the `insertionSort` is called as follows:

```c
int array[5] = {5, 1, 9, 3, 7};
...
insertionSort(array, 5);
```

Demonstrate your knowledge of arrays by listing the value of each element of the array `a` after every pass of the outer for loop.
3. The value of the mathematical constant $e$ can be calculated from the following infinite series:

$$e = \sum_{i=0}^{\infty} \frac{1}{i!} = \frac{1}{1} + \frac{1}{1} + \frac{1}{2} + \frac{1}{6} + \cdots$$

*Note:* $i!$ is defined to be $i$ factorial.

Define a function called `eCompute` with the following function prototype:

```c
double eCompute(int n);
```

The function should compute and return the sum of the first $n$ terms of the series above.
4. Consider the following program:

```c
#include "stdafx.h"

void processOneLine(int i);

const int height = 5;
int i;

int main(int argc, char* argv[]) {
    for (i = 1; i <= height; i++)
        processOneLine(i);
    return 0;
} // end function main

void processOneLine(int n) {
    for (i = 1; i <= height - n; i++)
        printf(" ");
    for (i = 1; i <= 2 * n - 1; i++)
        printf("*");
    printf("\n");
} // end function processOneLine
```

The programmer who wrote this code is trying to write a program that produces a triangle of asterisks ('*'); specifically, the following program output is desired:

```
*
***
*****
*******
*********
```

(a) The program does not work correctly. **What output is actually produced?**

(b) The program is actually very close to being correct. **Fix it by adding one line of code.**

Just indicate what line is needed, and where it should be inserted – nothing else needs to be changed.