

Exam 2 Review

COP 3223 Section 1

Questions 1 and 2

```
#include <stdio.h>

void function1(int array[], int length);

int main() {
    int values[] = {2, 5, 1, 0, 3, 4};
    function1(values, 6);
    return 0;
}

void function1(int array[], int length) {

    int temp[length], i;

    for (i=0; i<length; i++)
        temp[i] = array[array[i]];

    for (i=0; i<length; i++)
        array[i] = temp[i];

    for (i=0; i<length; i++)
        printf("%d ", array[i]);
}
```

1) Let the array values be an integer array of size 6 storing the following values (in this order): 2, 5, 1, 0, 3, 4. What is the output of running the function call `function1(values, 6)`?

- A) 2 5 1 0 3 4 B) 0 1 2 3 4 5 C) 5 4 3 2 1 0
 D) 1 4 5 2 0 3 E) NOTA

2) In order for `function1` NOT to cause an array out of bounds error, which of the following restrictions on the contents of the input array would suffice?

- A) Must be in between 0 and 5. B) Must be in between 0 and 6.
 C) Must be in between 0 and length-1.
 D) Must be in between 0 and length.
 E) Must be in between 1 and length.

Questions 3-6

```
#include <stdio.h>

int function2(int a, int b);

int main() {
    int a = 1, b = 2;
    b = function2(a,b);
    printf("a = %d, b = %d\n", a, b);
    a = function2(b,a);
    printf("a = %d, b = %d\n", a, b);

    return 0;
}

int function2(int a, int b) {
    int temp = a+b;
    a = 3*temp - (2*b)%15;
    b = a%temp;
    printf("a = %d, b = %d\n", a, b);
    return 2*a - b;
}
```

3) What is the first line of output produced by this program?

A) a = 2, b = 5 **B) a = 5, b = 2** C) a = 5, b = 1 D) a = 5, b = E) a = 3, b = 5

4) What is the second line of output produced by this program?

A) a = 1, b = 8 B) a = 2, b = 8 C) a = 8, b = 1 D) a = 8, b = 2 E) a = 8, b = 3

5) What is the third line of output produced by this program?

A) a = 26, b = 8 B) a = 8, b = 26 C) a = 7, b = 25 **D) a = 25, b = 7** E) a = 8, b = 7

6) What is the fourth line of output produced by this program?

A) a = 25, b = 8 B) a = 8, b = 43 C) a = 7, b = 43 D) a = 43, b = 7 **E) a = 43, b = 8**

Questions 7-10

```
#include <stdio.h>

int function3(int *a, int *b);

int main() {
    int a = 1, b = 2;
    b = function3(&a,&b);
    printf("a = %d, b = %d\n", a, b);
    a = function3(&b,&a);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}

int function3(int *a, int *b) {
    int temp = (*a)+(*b);
    *a = 3*temp - (2*(*b))%15;
    *b = (*a)%temp;
    printf("a = %d, b = %d\n", *a, *b);
    return 2*(*a) - (*b);
}
```

7) What is the first line of output produced by this program?

A) a = 2, b = 5 **B) a = 5, b = 2** C) a = 5, b = 1 D) a = 5, b = 3 E) a = 3, b = 5

8) What is the second line of output produced by this program?

A) a = 1, b = 8 B) a = 2, b = 8 **C) a = 5, b = 8** D) a = 8, b = 2 E) a = 8, b = 5

9) What is the third line of output produced by this program?

A) a = 3, b = 2 **B) a = 29, b = 3** C) a = 3, b = 26 D) a = 26, b = 3 E) a = 26, b = 8

10) What is the fourth line of output produced by this program?

A) a = 25, b = 55 B) a = 26, b = 55 C) a = 29, b = 55 D) a = 55, b = 26 **E) a = 55, b = 29**

Free Response 1

(20pts) Write a program that computes the average temperature for one month, averaged over all days of the month and all hours in a day. Store the data in a two-dimensional array named `temp_readings` where the rows of the array represent the days in a month (assume 30 days in a month) and the columns represent the hours in a day.

Assume the data is stored in a text file called `readings.txt`. In your program, read in the data from this text file into your array before you perform any computations. The format for the text file is simply a 30 by 24 matrix of doubles.

```

#include <stdio.h>

// define constants to be used through out the program
#define DAYS 30
#define HOURS 24

void main()
{
    // Declare the array according to specification of the array size
    // and input file (type of double).
    double temp_readings[DAYS][HOURS];

    // open input file
    FILE *inputFile = fopen("readings.txt", "r");

    // Check if the file exists
    if (inputFile == NULL)
    {
        printf ("No input file found\n");
        system ("pause");
        exit(1);
    }
}

```

```

int d, h;
double average = 0;

// Assuming the input file contains 30 lines for 30 days
// and each line contains 24 values for 24 hours in a day.

// for each day
for (d = 0; d < DAYS; d++)
{
    // for each hour in a day
    for (h = 0; h < HOURS; h++)
    {
        // read data into array before computation
        fscanf (inputFile, "%f", &temp_readings[d][h]);

        // add this reading to the average, which doubles as an accumulative sum for now.
        average += temp_readings[d][h];
    }
}

// compute the average
average /= DAYS*HOURS;

// output the result to the user
printf ("Average temperature for the month is %.2f\n", average);

fclose (inputFile);
system ("pause");
}

```

Free Response 2

(20pts) Write the following functions:

- A function that returns the largest element of an array
- A function that returns the average of all the elements in an array.
- A function that finds the number of positive elements in an array and returns this number as a parameter of the function (passing by reference). How would you call this function?

For these functions, assume that *a* and *n* are parameters where *a* is an array of int values and *n* is the length of the array. NOTE: for part c, your function does not return anything because you are passing by reference, thus, you will need another parameter in your function for the number of positive elements. Call this parameter *pos_num*.

```
#include <limits.h>

// This function worth 6 points
int findLargest (int a[], int n)
{
    int i;
    int largest = INT_MIN; //set to the smallest possible integer value in C

    // find the largest integer in the array
    for (i = 0; i < n; i++)
    {
        // If this element is larger than the largest found so far, replace the largest.
        if (a[i] > largest)
        {
            largest = a[i];
        }
    }

    // return the largest value
    return largest;
}
```

```
// This function worth 6 points
double findAverage (int a[], int n)
{
    int i;
    double average = 0;

    // First add all the numbers
    for (i = 0; i < n; i++)
    {
        average += a[i];
    }

    // Then divided by the number of the elements to find the average
    average /= n;

    // return the average value
    return average;
}
```

```
// This function worth 6 points
void countPositive (int a[], int n, int *pos_num)
{
    *pos_num = 0;
    int i;

    for (i = 0; i < n; i++)
    {
        // If this element is larger than zero, then the element is a positive number.
        if (a[i] > 0 )
        {
            // Add one to the number of positive integers found.
            (*pos_num)++;
        }
    }
}

• // 2 points
• countPositive (array, arraySize, &positiveCount);
```