**CSCD 240 Fall 2020 Name: Ian Kaiserman**

**Quiz 2:**

**Short Questions:**

**Question 1:** What is wrong with the following array initialization? How can you fix it?

int y[2] = { 1, 2, 3, 4, 5};

**The number of elements is larger than the initialized size. The methods for fixing it are either changing the size declaration from 2 to 5 or removing elements until there are only 2 left.**

**Question 2:** The following array needs to be sorted. The sort function needs the number of elements (here, **n**) in the array. How will you find ‘**n**’?

int num[] = {9, 7, 8, 6, 4, 5, 3, 1, 2, 0};

sort(num, n);

**The number of elements can be found by computing sizeof(num)/sizeof(int)**

**Question 3:** In the following two dimensional array ‘x’, what is the number of row and number of column?

char x[5][6];

**The number of rows is 5 and the number of columns is 6**

**Question 4:** What is wrong with the following program section?

int i , j;

int x[12][10];

for ( i = 0; i < 13; i++){

for ( j = 0; j < = 10; j++){

x[i][j] = i+j;

}

}

**The initialization of array x gives it a column amount of 12, but that means the index goes from 0 – 11, meaning that the very last iteration of both for loops is going to cause issues. The numbers need to be changed so that 11 and 9 are the last numbers processed for i and j respectively to avoid an out of bounds error.**

**Question: 5** What is the number of rows in the following 2-dimensional array?

int data[][COLUMNS] = { {15, 2, 9, 4,0}, {-1, 4, 2, 3, 7}, {2, 3, 6, 1, 0}};

**The implicit number of rows is 3.**

**Question 6:** Consider the following two dimensional array. What is the value of **array[2][2]**?

int array[3][4] ={ 1, 2, 3, 4, 5, 6, 7, 8, 9};

**The value of array[2][2] is 0 by default because it wasn’t explicitly defined**

**Question 7:** What will be printed by the following function?

#include <stdio.h>

int main(){

float x1 = 0.0005, x2 = -12.88;

printf("x1=%.2f x2=%.3f\n", x1, x2);

return 0;

}

**x1=0.00 x2=-12.880**

**Question 8**: What will be printed by the following program?

#include <stdio.h>

int main(){

int i = 7, j;

printf(“input a number: “);

scanf(“%d”, &j); // suppose you typed 5 from the keyboard

int ret = ( i == j);

printf(“%d”, ret);

return 0;

}

**Input a number: 5**

**0**

**Question 9:** What will be printed by the following program?

#include <stdio.h>

int main(){

int total = 23;

int count = 10;

double average = total/count;

printf(“Average is: %f\n”, average);

return 0;

}

**Average is: 2.000000**

**Question 10:** What will be the final values of the array elements in the following program?

#include <stdio.h>

void update (int value);

int main(){

int i;

int grades[5] = { 80, 80, 90, 60, 40 };

update(grades[2]);

for (i = 0; i < 5; i++)

printf("%d ", grades[i]);

printf("\n");

return 0;

}

void update(int value){

value += 10;

}

**80**

**80**

**90**

**60**

**40**

**Identify whether the following statements are true or false.**

**Question 11:** The following function declaration is correct.

void readArrayElements( int a[][], int nRows, int nCols);

**True**

**Question 12:** Any function written to receive an array argument is given the address of the first element of the array passed to it.

**True**

**Question 13:** The following two expressions are equivalent.

char name[] = “Robin”;

char name[5] = “Robin”;

**False**

**Question 14:** Arrays globally defined are implicitly initialized.

**True**

**Question 15:** In array ‘**a**’, array index ranges from 0 to 10.

**int a[10];**

**False**

**Question 16:** A multi-dimensional array is translated by the compiler into a single dimensional array.

**False**

**Question 17:** The following expression is true.

**sizeof(char) > sizeof(int)**

**False**

**Question 18:** A character can be treated as either an integer or a character, depending on its use.

**True**

**Question 19: fgets()** replaces the operator generated new line character ‘\n’ with a NULL-terminated character ‘\0’.

**False**

**Question 20**: A function declaration is about writing the function body and completing the function.

**False**