Question 1:

a. False because the y is storing the address of x not the value of x
b. True because by making *y = 5, it will change the address to 5 where x is located

Question 4

a. I flipped *array2 and array1 when they start inputting values into the arrays

Question 6

b. Corrections on test paper.

Question 7

a. The pointer is pointing to a read-only array. Change the void setData(const int* data) to void setData(int* data).
b. The code is finding NUMITEMS and replacing it with a value of 5.

Exam 2

CS 2301

4/10/23

Name Jonathan King

Instructions:

1. Record your name on the line above.
2. Write your name **on the back** of the last page
3. Show your work to receive partial credit

Question	Possible points	Score
1	10	6
2	5	5
3	5	5
4	20	15
5	15	15
6	25	22
7	20	5+2
Total	100	~~73~~ 75

These boxes are for grading

1. (2 pts each) Label each statement about the following lines of code as true or fales.

```
int x = 0;

int* y = &x;
```

true The variable y stores the value of 0.

false *y = 5; will change the value stored in x.

false the variable y can be used to point to a float without causing confusion.

false the variable x can be used to point to an integer.

true The variable y stores the memory address of x.

2. (5 pts) What will this code display? (Draw the variables for partial credit)

```
float sample1 = 8;
float* sample2 = &sample1;
sample1 = 2;
printf("sample1: %d, sample2: %d \n", sample1, *sample2);
```

sample1: 2, sample2: 2

mem	Val	
Sample1	8	→ 2
Sample2	Sample1	→ 2

3. (5 pts) Given the following definition, what is the value of b[1][0]?

```
int b[2][2] = {{2, 6}, {9, 4}};
```

b[1][0] = 9

$$\begin{bmatrix} 2 & 6 \\ 9 & 4 \end{bmatrix}$$

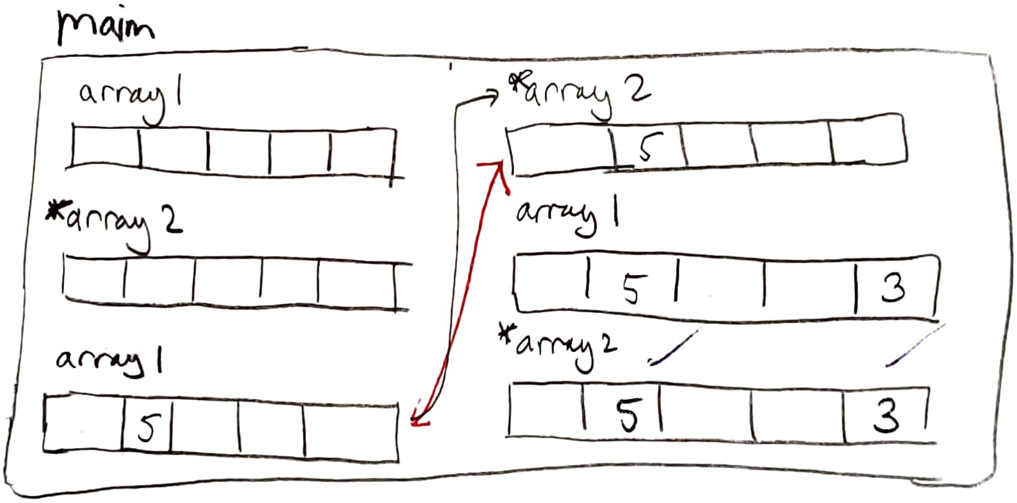
1: 6 2: 5 3: 5

4. (20 pts) Consider code shown below.

```
1   int main(void){
2       int array1[5];
3       int* array2;
4       array2 = array1;
5       array2[1] = 5;
6       array1[4] = 3;
7       // Diagram here
    }
```

4A. (15 pts) Draw a diagram of the variables on the stack when the program reaches line 7.

+10



-5

4B. (5 pts) Write a few lines of code to set all values of array1 to equal 8.

+5

```
for (int i=0; i<5; i++)
{
    a[i] = 8;

}
```

4: 15

Use this code, `primeNumbers.c`, to answer questions 5 and 6.

```
1    #include <stdio.h>
2    #include <stdlib.h>
3
4    int is_prime(int);
5
6    int main(int argc, char* argv[]){
7        int to_print = atoi(argv[2]); 5
8        int num_to_check = atoi(argv[1]); 7
9        int counter = 0;
10                               5
11       while (counter < to_print){
12           if (is_prime(num_to_check) == 1){
13               printf("%d ", num_to_check);
14               counter = counter + 1;
15           }
16           num_to_check = num_to_check + 1;
17       }
18   }
19                        7
20   int is_prime(int value){
21
22       int val_check = 2;
23      // Diagram here for the first call of is_prime()
24       while (val_check < value){
25           if (value % val_check == 0){
26               return 0;
27           }
28           val_check++;
29       }
30       return 1;
31   }
```

5. (15 pts) Briefly explain each section of code (similar to the content of a comment)

Lines 1-2:

imports C libraries

Line 4:

initiallzes a function for main to call to

Line 7:

sets the 3rd argument to an integer and sets it as to_print

Line 12-15:

if num_to_check is prime, add 1 to counter

Line 24-29:

while value is greater than 2, check if value is prime by dividing value with all numbers from 2-(value-1)

5: 15

Question 6 (25 pts)

6A. (5 pts) What is the final output of the program if the program is run using the following command?

```
gcc primeNumbers.c -o prime.exe
./prime.exe 7 5
```

7 13 17 23 29

6B. (20 pts) Draw all variables on the stack the first time the program reaches line 23.

main	is_prime
to_print = 5	value = num_to_check = 7
num_to_check = 7	~~val_check = 2~~ doesn't reach
counter = 0	

argc = 3
argv[1] = 7
argv[2] = 5

X -1

argv +argc -2

6: 22

Use this code to answer #7

```
1    #include <stdio.h>
2    #define NUMITEMS 5
3
4    void setData(int*);
5
6    int main(void){
7
8        int values[NUMITEMS];
9        setData(values);
10
11       for (int j = 0; j < NUMITEMS; j++){
12           printf("%3d\n", values[j]);
13       }
14       return 0;
15   }
16
17   void setData(const int* data){
18       for(int i = 0; i < NUMITEMS; i ++){
19           data[i] = i;
20       }
21   }
```

Error message:

```
array_data.c: In function 'setData':
array_data.c:19:17: error: assignment of read-only location '*(data +
     (sizetype)((long long unsigned int)i * 4))'
         data[i] = i;
```

7. Use the code to answer the following questions.

7A. (10 pts) There is a problem in the code above. Explain the error and propose a solution. Your solution must still involve the use of the keyword const

The pointer int*data is not being addressed to anything.

Change to void set Data (const int * pData)

7B. (5 pts) What is the purpose of line 2?

Sets the global variable NUMITEMS as 5

7C. (5 pts) After fixing the problem, what is the output of this program?

0
1
2
3
4

7: 5