SE185: Problem Solving in Software Engineering
Quiz # 7 (200 points)

Name: Adam Jennissen	Name:

Answer the following questions and make a pdf file that includes the **source code, sample inputs, and outputs**. You must submit the **pdf file and all of the .c files** on Canvas for full credit. Do not forget to add your group partner name on the pdf file and the source codes.

1. (50 points) Write a complete C program that declares an integer called num and initialize it to 5. Create an integer pointer variable called myPtr that stores the memory address of num. Print the memory addresses of num and myPtr, the values stored in num and myPtr, and the value that myPtr points to in this format:

num is stored at: ____

myPtr is stored at: ____

num holds the value: ____

myPtr holds the value: ____

myPtr points to this value: ____

Hint: The value that num holds and value that myPtr points to are equal.

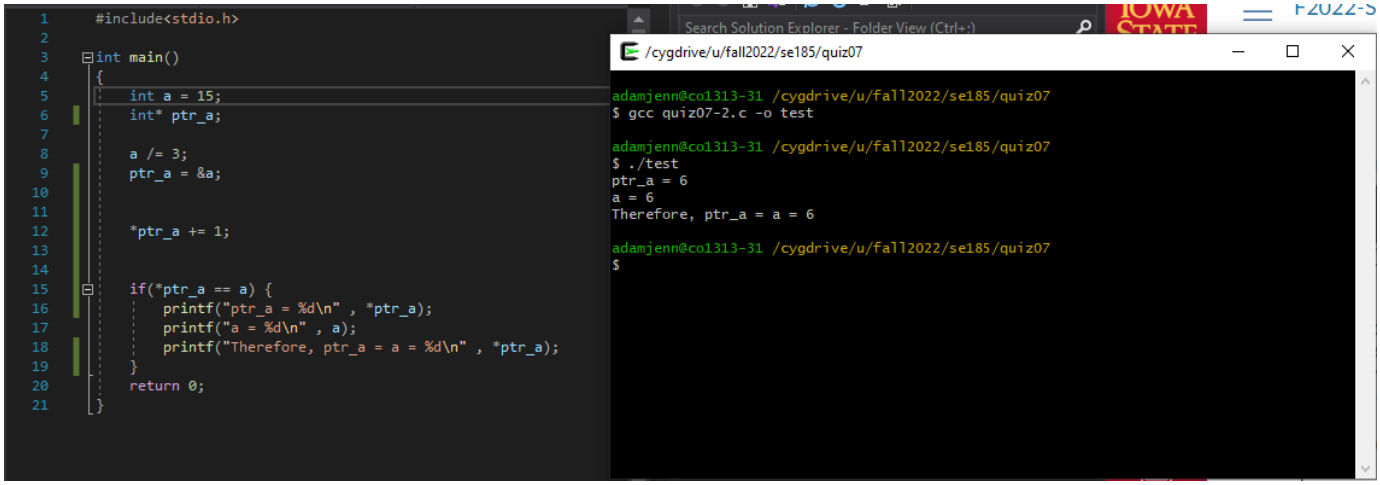
```c
#include <stdio.h>

int main(void)
{
    int num = 5;
    int* myPtr;
    *myPtr = num;

    printf("num is stored at: %d\n", &num);
    printf("myPtr is stored at: %d\n", &myPtr);
    printf("num holds the value: %d\n", num);
    printf("myPtr holds the value: %d\n", myPtr);
    printf("myPtr points to this value: %d\n", *myPtr);
    return 0;
}
```

```
/cygdrive/u/fall2022/se185/quiz07

adamjenn@co1313-31 /cygdrive/u/fall2022/se185/quiz07
$ ./test
num is stored at: -13268
myPtr is stored at: -13280
num holds the value: 5
myPtr holds the value: -12816
myPtr points to this value: 5

adamjenn@co1313-31 /cygdrive/u/fall2022/se185/quiz07
$
```

2. (50 points) Re-implement the following code by defining int copy_a as an integer pointer variable called ptr_a. Keep int as an integer variable.

```c
#include<stdio.h>

int main() {
   int a = 15;
   int copy_a = a;
   a /= 3;
```

```
    copy_a = a;

    copy_a++;
    a = copy_a;

    if(copy_a == a) {
        printf("Copy_a = %d\n" , copy_a);
        printf("a = %d\n" , a);
        printf("Therefore, copy_a = a = %d\n" , copy_a);
    }

    return 0;
}
```



Inputs and outputs format:

```
Copy_a = 6
a = 6
Therefore, copy_a = a = 6
```

3. (100 points) Write a complete C program that ask users to enter midterm 1 exam score for 30 students. Your program then calculates following exam statistics and print the result.

(a) Midterm 1 exam average

(b) Maximum score

(c) Minimum score

(d) Number of students fail (<60)

(e) Number of students got A (93+)

Your program must meet the following requirements:

1. Store the user inputs (midterm 1 exam scores) to an array named **midterm1Score**

2. You must use a user defined function named **examStat** to calculate the exam statistics, and save the result to an array named **result**.
    - When you call the function, you must **pass four arguments** including **two arrays and the size of the two arrays**.
    - Calculate the exam statistics (mentioned above), and **save the result to an array**.
3. Print the exam statistics from the array named **result**.

```c
#include <stdio.h>

void examStat(int scores[], int score_size, int results[], int result_size);

int main(void)
{
    const int num_scores = 30;
    const int size_result = 5;
    int midterm1Score[num_scores];
    int result[size_result];
    int i;

    for (i = 0; i < num_scores; i++)
    {
        printf("Score %d = ", (i + 1));
        scanf(" %d", &midterm1Score[i]);
    }

    examStat(midterm1Score, num_scores, result, size_result);

    printf("\nThe average score is: %d\n", result[0]);
    printf("The maximum score is: %d\n", result[1]);
    printf("The minimum score is: %d\n", result[2]);
    printf("The number of students who failed is: %d\n", result[3]);
    printf("The number of students who got an A is: %d\n", result[4]);

    return 0;
}

void examStat(int scores[], int score_size, int results[], int result_size)
{
    int i;
    int temp = 0;

    for (i = 0; i < score_size; i++)
    {
        temp += scores[i];
    }
    results[0] = (temp / score_size);

    temp = scores[0];
    for (i = 0; i < score_size; i++)
    {
        if (scores[i] > temp)
            temp = scores[i];
    }
    results[1] = temp;

    temp = scores[0];
    for (i = 0; i < score_size; i++)
    {
        if (scores[i] < temp)
            temp = scores[i];
    }
    results[2] = temp;

    temp = 0;
    for (i = 0; i < score_size; i++)
    {
        if (scores[i] < 60)
            temp++;
    }
```

```
/cygdrive/u/fall2022/se185/quiz07
adamjenn@co1313-31 /cygdrive/u/fall2022/se185/qu
$ gcc quiz07-3.c -o test -Wall

adamjenn@co1313-31 /cygdrive/u/fall2022/se185/qu
$ ./test
Score 1 = 43
Score 2 = 89
Score 3 = 58
Score 4 = 45
Score 5 = 69
Score 6 = 89
Score 7 = 58
Score 8 = 98
Score 9 = 95
Score 10 = 97
Score 11 = 96
Score 12 = 88
Score 13 = 85
Score 14 = 75
Score 15 = 25
Score 16 = 35
Score 17 = 69
Score 18 = 68
Score 19 = 73
Score 20 = 56
Score 21 = 11
Score 22 = 68
Score 23 = 25
Score 24 = 58
Score 25 = 54
Score 26 = 15
Score 27 = 28
Score 28 = 95
Score 29 = 68
Score 30 = 73

The average score is: 63
The maximum score is: 98
The minimum score is: 11
The number of students who failed is: 13
The number of students who got an A is: 5

adamjenn@co1313-31 /cygdrive/u/fall2022/se185/qu
$
```