Name: ______________________________ CS 232

Date: ______________________________ Pointer Practice

For each block of code, determine if it is syntactically correct and if it is legal code (e.g., no dereferencing of invalid pointers). If it is both syntactically correct and legal, answer the question about the value(s) of certain expressions and/or variables at the end of the code.

1.

```
int x=1, y=2;
int *pOne, *pTwo;

pOne = x;
pTwo = y;

*pOne = *pTwo;
```

What are x and y?

2.

```
double c=3.0, d=4.5;
double *pOne, *pTwo;
pOne = &c;
pTwo = &d;

*pTwo = c;
*pOne –= 1.0;
d += 2.2;
```

What are c, d, *pOne, and *pTwo?

3.

```
double x=2.1, y=3.5;
double *pOne, *pTwo;

y –= x;
*pOne = 4.0;
*pTwo = *pOne * 2.0;
```

What are x, y, *pOne, and *pTwo?

4.

```
double x=2.1, y=3.5, *pOne;
pOne = &y;

*&x += 1.9;
*pOne += x;
```

What are x, y, and *pOne?

5.

```
int x=1, y=2;
int *pOne, pTwo, pThree;

pOne = &x;
pTwo = &y;

pThree = pOne;
pOne = pTwo;
*pThree = *pOne;
```

What are x, y, *pOne, *pTwo, and *pThree?

6.

```
int y=7, z=11;
int *pOne = &y, *pTwo = &z;

pOne += 2;
z += *pOne;
```

What are y, z, *pOne, and *pTwo?

Pointers and Functions

1.

```
void getPowers(int x, int xSqr, int xCube){
    xSqr = x * x;
    xCube = x * xSqr;
}
int main(){
    int x, xSqr, xCube;
    x = 3;
    getPowers(x, xSqr, xCube);
```

What are x, xSqr, xCube?

2.

```
void getPowers(int x, int *pxSqr, int *pxCube){
    *pxSqr = x * x;
    *pxCube = x * *pxSqr;
}
int main(){
    int x, xSqr, xCube;
    x = 3;
    getPowers(x, xSqr, xCube);
```

What are x, xSqr, xCube?

3.

```
void getPowers(int x, int *pxSqr, int *pxCube){
    *pxSqr = x * x;
    *pxCube = *pxSqr * x;
}
int main(){
    int x, *pxSqr, *pxCube;
    x = 3;
    getPowers(x, pxSqr, pxCube);
```

What are x, *pxSqr, *pxCube?

4.

```
void getPowers(int x, int *pxSqr, int *pxCube){
    *pxSqr = x * x;
    *pxCube = *pxSqr * x;
}
int main(){
    int x, xSqr, xCube;
    x = 3;
    getPowers(x, &xSqr, &xCube);
```

What are x, xSqr, xCube?

Pointers, Arrays, and Functions

1. The nth Fibonacci number is defined as the sum of the previous two Fibonacci numbers. The first and second Fibonacci numbers (which are needed to get the recurrence started) are 0 and 1, respectively.

Write a function that takes a single argument, an int* that corresponds to an array of size (at least) 3. The precondition of the function is that the first two elements of the array are the (n-2) and (n-1) Fibonacci numbers. The function should compute the nth Fibonacci number and place it in the third element of the array.

2. Fill in the following main function so that it fills the first n Fibonacci numbers into the fib array. Use the function you defined above to compute each number.

```
int main(){
    const int n = 25;
    int fib[n];
    int i;
    fib[0] = 0;
    fib[1] = 1;

    /* fill this in */










}
```