

**Question 1.** [2 points] Given the following array, what is the value of `quiz[5]`?

```
float quiz[5] = {88.9, 56.5, 92.1, 78.0, 68.9};
```

A. Can't be predicted

B. 0

C. 68.9

D. 88.9

**Question 2.** [3 points] Given the following code snippet, what is the expected output of the `printf` statement?

```
int deca[3] = {12, 2, 0};  
deca[2] = deca[0] + deca[1];  
printf("%i", deca[2]);
```

A. Can't be predicted

B. 0

C. 12

D. 14

**Question 3.** [3 points] From the choices below, circle *all* valid function prototypes (there may be more than one).

A. `int sumNum(int x, y, z);`

B. `float avg(float exam1, int exam2);`

C. `void drawSq(int squareHeight);`

D. `printLine(size);`

**Question 4.** [6 points] Add code to the following program just below the TODO comment so that the loop containing the printf statement will print the output 17 42 121. Use assignment statements to set the values, i.e. do **not** use scanf to obtain the inputs.

```
#include <stdio.h>

int main(void) {
    int arr[10];
    int m, n;

    // TODO
    m = 0;
    n = 2;
    arr[0] = 17;
    arr[1] = 42;
    arr[2] = 121;

    for (int i = m; i <= n; i++) {
        printf("%i ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

**Question 5.** [6 points] What output is printed by the following program (which begins on the left and continues on the right)?

<pre>#include &lt;stdio.h&gt;  int findMin(int a, int b);  int main(void) {     int x = 6, y = 7;     int min = 0;     findMin(x, y);     printf("%i\n", min);     return 0; }</pre>	<pre>int findMin(int a, int b) {     int min;     if (a &lt; b) {         min = a;     } else {         min = b;     }     return min; }</pre>
--	--

note:  
return  
value  
not  
used

0

**Question 6.** [8 points] What output is printed by the following program (which begins on the left and continues on the right)?

```
#include <stdio.h>

#define N 6

int multiply(int x, int y) {
    return x * y;
}
```

```
int main(void) {
    int a[N] = {2, 4, 6, 8, 10, 12};
    int product = 0;
    printf("Start Here!\n");

    for (int i = 0; i < N; i++) {
        product = multiply(a[i], i+1);
        printf("Multiply %i: %i\n",
              i, product);
    }

    printf("All Done!\n");
    return 0;
}
```

Start Here!  
Multiply 0: 2  
Multiply 1: 8  
Multiply 2: 18  
Multiply 3: 32  
Multiply 4: 50  
Multiply 5: 72  
All Done!

For Questions 7–12, circle True or False.

**Question 7.** [2 points] True or False: A function can use a return statement to return more than one value.

**Question 8.** [2 points] True or False: All functions must return a value.

\* **Question 9.** [2 points] True or False: A function call may have more argument expressions than there are function parameter variables.

**Question 10.** [2 points] True or False: Variables defined in the body of the main function can be accessed by any other function in the program.

**Question 11.** [2 points] FALSE If a function with a non-void return type does not execute a return statement, then the function will automatically return the value 0.

**Question 12.** [2 points] It is legal to declare a variable with the same name in the bodies of two different functions. TRUE

\* Unless the function allows a variable argument list, which is a language feature we have not discussed.