

6.1903 Quiz 1 (9/28/2026)

ANSWER KEY

Name:

Kerberos:

Enter your answers in the boxes provided. Additional scratch paper is provided but will not be considered in grading.

1.

You'll recall from lab 1 you wrote the function `pinWrite(int pin_num, int value)` that:

- Sets bit `pin_num` in the 32 bit word stored at address `GPIO_OUT_ADDR` to be equal to an input value (with `value` being limited to either 0 or 1).
- Returns nothing (note the void return type).

Maintaining consistency with this framework, write a function called `togglePin` that takes in a `pin_num` and flips ($0 \rightarrow 1$ or $1 \rightarrow 0$) the appropriate bit specified by `pin_num` from its current value leaving all other bits unchanged. This is to be done in **one line using bitwise operations only, (no if/else branch logic or ternary operations)**.

```
void togglePin(int pin_num){
    int *output_val = (int*) GPIO_OUT_ADDR;
    *output_val ^= (1 << pin_num);
    //or
    *output_val = *output_val ^ (1 << pin_num);
}
```

2.

(a) Here is an integer:

```
int8_t a = 14;
```

Negate the integer. Give your answer in 2's complement binary (8 bits).

Answer (binary): 0b11110010

(b) You have the following:

```
float a = 20.5;
// binary encoding: 0b01000001101001000000000000000000
```

Divide the float `a` by 2. Give your answer in hexadecimal

Answer (hexadecimal): 0x41240000

(binary: 01000001001001000000000000000000, as a float: 10.25)

3.

```
void f1(int * array, int len){
    for (int i = 0; i < len; i++){
        array[i] = array[i] - 2;
    }
}

int f2(int * array, int index_1, int index_2){
    return array[index_1] + *(array + index_2);
}

void app_main(){
    int a[10] = {10, 40, 65, 64, 88, 56, 39, 43, 23, 89};
    f1(a + 2, 8);
    int result_1 = f2(a, 2, 0);
    int result_2 = f2(a+3, 1, 3);
    int result_3 = f2(a+2, 3, 5);
    printf("calculation 1: %d\n", result_1);
    printf("calculation 2: %d\n", result_2);
    printf("calculation 3: %d\n", result_3);
}
```

(a) In the box below, fill in the largest possible value for `__UNKNOWN__` above so that `f1` cannot read or write past the end of the array `a`.

Answer: 8

(b) What values are printed for calculation 1, calculation 2, and calculation 3? Assume `__UNKNOWN__` is your answer from part (a).

Answers:

calculation 1: 73

calculation 2: 123

calculation 3: 95

a after **f1**: {10, 40, 63, 62, 86, 54, 37, 41, 21, 87}