# Advanced C Practice Worksheet

**🔧 Focus: Logic, Functions, Variable Scope, Data Types, and Control Flow**

## 🧩 Section A: Debug This Code

### 🔧 Q1. Function Return Problem

```
int compute(int a, int b){
    if(a > b)
        return;
    else
        return a - b;
}

int main(){
    int x = 10, y = 5;
    printf("Result: %d", compute(x, y));
}
```

### 🔧 Q2. Array Condition Confusion

```
int main(){
    int scores[3] = {80, 90, 100};
    if (scores == {80, 90, 100}){
        printf("All scores matched");
    }
    return 0;
}
```

### 🔧 Q3. Control Flow Misuse

```
int main(){
    int num = 5;

    if(num = 0){
        printf("Zero");
    }
    else if(num > 0){
        printf("Positive");
    }
    else{
        printf("Negative");
    }
    return 0;
}
```

---

### 🔧 Q4. Unexpected Output with Floats

```
float divide(int a, int b){
    return a / b;
}

int main(){
    printf("Division: %.2f", divide(5, 2));
    return 0;
}
```

---

## 🧠 Section B: Short Answer Questions

### 📝 Q1. What is the difference between a function declaration and a function definition in C?

### 📝 Q2. Why is comparing arrays directly with `==` invalid in C, and what is the alternative?

### 📝 Q3. Explain why `5 / 2` gives `2` in C but `2.5` in Python.

### 📝 Q4. How does passing by value in C affect functions that modify variables?

### 📝 Q5. What would happen if a `float` is passed to a function expecting an `int`?

---

## ❓ Section C: Multiple Choice Questions

### 🔘 Q1. What happens if you return from a `void` function in C?

a) Compiler error
b) It returns `0`
c) It works if `return;` is used
d) Program crashes

---

### 🔘 Q2. What best describes a block-scoped variable?

a) Can be used in any file
b) Shared across functions
c) Only available within its defining `{}`
d) Needs to be declared as `global`

---

### 🔘 Q3. What is the size of a `double` on a typical 64-bit compiler?

a) 4 bytes
b) 6 bytes
c) 8 bytes
d) 10 bytes

---

### 🔘 Q4. What will be the result of `7 / 2` in C?

a) 3.5
b) 4
c) 3
d) 3.0

---

## 💻 Section D: Code Writing Challenges (No Recursion or Array Parameters)

### 🧑‍💻 Q1. Write a function `int rotate_digits(int num)` that:

- Takes a 3-digit number (e.g., `123`)
- Rotates its digits to the left (`231`)
- Returns the rotated number
- Do **not** use arrays or strings

---

### 🧑‍💻 Q2. Implement a function `int reverse_number(int num)` that:

- Reverses any positive number using loops only
- Then checks in `main()` whether the number is a palindrome
- Do **not** use arrays, strings, or recursion

---

### 🧑‍💻 Q3. Write a function `float compute_average(int total, int count)` that:

- Returns the average (as float)
- Take user input for `total` and `count` in `main()`
- Display result with 2 decimal precision
- Print the size of `int` and `float` in bytes

---

## 🧑‍💻 Q4. Write a program that:

- Accepts a character from the user
- Checks whether it's a vowel using a loop and `char` comparison
- Do **not** use arrays for vowels
- Hint: Use multiple `if` or `switch`

---

## 🧑‍💻 Q5. Create a program that:

- Prints the **size** and **maximum value** of:
  - `int`, `float`, `double`, `char`
- Use `<limits.h>` and `<float.h>` for reference

---