

Programming in C and C++ - Practice Sheet 3

First Name: _____

Last Name: _____

Matriculation Number: _____

Instructions:

1. Read all questions before starting.
2. Write your name above immediately.
3. Write clearly. C/C++ are case-sensitive.
4. Indent your code properly.
5. No books, notes, or electronic devices allowed.
6. You may use the back of pages for extra space.
7. You have 2 hours to complete this test.
8. Cheating results in immediate failure.
9. By signing below, you confirm understanding of all rules.

Signature: _____

%	Grade	%	Grade	%	Grade	%	Grade
0.00 - 39.49	5.0	39.50 - 44.49	4.7	44.50 - 49.49	4.3	49.50 - 54.49	4.0
54.50 - 59.49	3.7	59.50 - 64.49	3.3	64.50 - 69.49	3.0	69.50 - 74.49	2.7
74.50 - 79.49	2.3	79.50 - 84.49	2.0	84.50 - 89.49	1.7	89.50 - 94.49	1.3
	94.50 - 100.00				1.0		

Problem 1: String Case Conversion (3 points)

Language: C

Write `void to_uppercase(char *str)` that converts all lowercase letters to uppercase without using `string.h` functions.

Problem 2: Matrix Symmetry Check (4 points)

Language: C

Write a function that checks if a square matrix is symmetric:

```
1 int is_symmetric(int **matrix, int n) {  
2     // Return 1 if symmetric, 0 otherwise  
3 }
```

Problem 3: Pointer Arithmetic (3 points)

Language: C

What is the output?

```
1 #include <stdio.h>  
2 int main() {  
3     int arr[] = {1, 2, 3, 4, 5};  
4     int *p = arr;  
5     printf("%d\n", *p++);  
6     printf("%d\n", (*p)++);  
7     printf("%d\n", *++p);  
8     printf("%d\n", ++*p);  
9     return 0;  
10 }
```

Problem 4: Recursive GCD (5 points)

Language: C

Implement recursive GCD using Euclid's algorithm:

```
1 int gcd(int a, int b) {  
2     // Your code here  
3 }
```

Problem 5: Dynamic String Concatenation (4 points)

Language: C

Write a function that concatenates two strings and returns a new dynamically allocated string:

```
1 char* concatenate(const char* s1, const char* s2) {  
2     // Your code here  
3 }
```

Remember to allocate enough memory and free it later.

Problem 6: Stack Implementation (6 points)

Language: C

Implement a stack using linked list with these operations:

- `push(int value)`: Add to top
- `int pop()`: Remove and return top
- `int peek()`: Return top without removal
- `int is_empty()`: Check if empty

Define the node structure and implement all functions.

Problem 7: Binary File Operations (3 points)

Language: C

Write a program that reads integers from binary file "data.bin" and prints them in reverse order using `fread()` and `fseek()`.

Problem 8: Bitwise Swap (2 points)

Language: C

Swap two integers using XOR without temporary variable:

```
1 void swap_bits(int *a, int *b) {  
2     // Your code here  
3 }
```

Problem 9: Date Class (5 points)

Language: C++

Design a `Date` class with:

- Private: day, month, year
- Constructor with validation
- Method to check leap year
- Method to add days to date
- Overloaded == and < operators

Problem 10: Copy Concepts (4 points)

Language: C++

Explain with examples:

1. Shallow copy vs Deep copy
2. When custom copy constructor is needed
3. Rule of Three in C++

Problem 11: Vehicle Inheritance (6 points)

Language: C++

Create inheritance hierarchy:

- **Vehicle**: brand, year
- **Car** (inherits **Vehicle**): numDoors
- **SportsCar** (inherits **Car**): topSpeed

Implement constructors and virtual `displayInfo()` method.

Problem 12: Subscript Operator (3 points)

Language: C++

Overload [] operator for **IntArray** class:

```
1 class IntArray {
2     int *data;
3     int size;
4 public:
5     int& operator[](int index); // Implement with bounds checking
6 };
```

Problem 13: File Exception Handling (2 points)

Language: C++

Write code that attempts to open a file and throws/catches a custom exception if opening fails.

Problem 14: Template Class (5 points)

Language: C++

Implement a template class `Pair<T1, T2>` that stores two values of potentially different types with getters and a swap method.