EECS-22: Advanced C Programming (Winter 2025)
Midterm Exam, Part 1 (02/10/25)

Prepared by: Dr. Halima Bouzidi (EECS-22 Instructor)

Full Name:	
Student ID:	
Lab session:	
Signature:	

- This is a closed-book, closed-notes exam.
- All electronic devices must be turned off. No devices are permitted.
- You have 50 minutes to complete the exam.
- The exam consists of 30 multiple-choice questions found on pages 2 to 5 (Make sure you have all the pages!)
- This page serves as the answer sheet for multiple-choice questions. Mark your answers clearly by filling in the circles on this page. Answers marked on other pages will not be considered.
- Each multiple-choice question has four (04) answer choices. Mark the corresponding circle(s) on this answer sheet.
- There are two types of multiple-choice questions:
  - Questions worth one point have only one correct answer.
  - Questions worth two points may have multiple correct answers (Select all that apply).
- For multiple-choice questions, each incorrect selection negates one correct selection. If you fail to select at least half of the correct answers, you will receive no points.
- In the provided program code, line numbers are for reference only and are not part of the code itself.
- No questions are allowed during the exam. If in doubt, write a note and clearly state all your assumptions.

Q#	a	b	c	d
1	○	○	○	○
2	○	○	○	○
3	○	○	○	○
4	○	○	○	○
5	○	○	○	○
6	○	○	○	○
7	○	○	○	○
8	○	○	○	○
9	○	○	○	○
10	○	○	○	○

Q#	a	b	c	d
11	○	○	○	○
12	○	○	○	○
13	○	○	○	○
14	○	○	○	○
15	○	○	○	○
16	○	○	○	○
17	○	○	○	○
18	○	○	○	○
19	○	○	○	○
20	○	○	○	○

Q#	a	b	c	d
21	○	○	○	○
22	○	○	○	○
23	○	○	○	○
24	○	○	○	○
25	○	○	○	○
26	○	○	○	○
27	○	○	○	○
28	○	○	○	○
29	○	○	○	○
30	○	○	○	○

1. **Which Linux command is used to display the current working directory?**
   (a) `pwd`
   (b) `whoami`
   (c) `cd`
   (d) `echo $pwd`

---

2. **What is the purpose of the `preprocessor` in C? (Select all that apply!)**
   (a) To define a new data type
   (b) To declare a variable
   (c) To define a macro
   (d) To include a library

---

3. **Which Linux shell command is used to list all files in a directory, including hidden files?**
   (a) `ls`
   (b) `ls -a`
   (c) `ls -l`
   (d) `ls -h`

---

4. **What is the purpose of structured program composition in C?**
   (a) To improve code readability and maintainability
   (b) To reduce program execution time
   (c) To increase memory usage
   (d) To eliminate the need for functions

---

5. **What is the purpose of a Makefile in C programming?**
   (a) To automate the compilation process
   (b) To debug the program
   (c) To define macros
   (d) To allocate dynamic memory

6. **What is the output of the following code?**

```c
#include <stdio.h>
#define SQUARE(x) ((x) * (x))
int main() {
    int a = 5;
    int b = a++;
    printf("%d", SQUARE(b));
    return 0;
}

```

   (a) 25
   (b) 30
   (c) 36
   (d) Undefined behavior.

---

7. **Which of the following storage classes in C has the shortest lifetime? (Select all that apply!)**
   (a) `auto`
   (b) `static`
   (c) `extern`
   (d) `register`

---

8. **What is the scope of a static variable declared inside a function in C?**
   (a) Limited to the function where it is declared.
   (b) Accessible throughout the entire program.
   (c) Restricted to the file where it is declared.
   (d) Local to the block (i.e., inside {}) in which it is declared inside the function.

---

9. **What is the output of the following code?**

```c
#include <stdio.h>
int main() {
    int x = 10;
    printf("%d", x << 2);
    return 0;
}

```

   (a) 10
   (b) 20
   (c) 40
   (d) Undefined behavior.

---

10. **What is the purpose of the `continue` statement in a loop? (Select all that apply!)**
    (a) It terminates the loop.
    (b) It skips the current iteration and directly proceeds to the next iteration.
    (c) It exits the program.
    (d) It causes the loop to jump to its next iteration.

---

11. **What is the output of the following code?**

```c
#include <stdio.h>
int main() {
    int i = 0;
    while (i < 3) {
        static int x = 0;
        x++;
        printf("%d ", x);
        i++;
```

```
        }
        return 0;
    }
```

(a) 0 1 2
(b) 1 2 3
(c) 1 1 1
(d) 0 0 0

---

12. **What is the output of the following code?**

```
    #include <stdio.h>
    int main() {
        const int a = 5, b = 10;
        if (a = b) printf("Equal");
        else printf("Not Equal");
        return 0;
    }
```

(a) Equal
(b) Not Equal
(c) Compilation error
(d) Undefined behavior.

---

13. **Which of the following statements about recursion in C are true? (Select all that apply!)**

(a) Recursion requires a base case to terminate.
(b) Recursion can be used with functions.
(c) Recursion uses less memory than iteration.
(d) Recursion is generally slower than iteration due to function call overhead.

---

14. **What is the output of the following code?**

```
    #include <stdio.h>
    int main() {
        int arr[] = {1, 2, 3, 4, 5};
        printf("%d", arr[2]);
        return 0;
    }
```

(a) 1
(b) 2
(c) 3
(d) 4

---

15. **Which of the following statements about debugging with gdb and macros in C are true? (Select all that apply!)**

(a) The gdb command '`break`' sets a breakpoint at a specific line number or function.
(b) Macros defined with '`#define`' can be debugged using gdb like regular functions.
(c) The gdb command '`print`' can be used to display the value of a variable during debugging.
(d) Macros are expanded by the preprocessor, so they cannot be stepped into during gdb debugging.

---

16. **What is the purpose of the `typedef` keyword?**

(a) To define a new data type (i.e., alias)
(b) To declare a variable
(c) To define a macro
(d) To include a library

---

17. **What is the output of the following code?**

```
    #include <stdio.h>
    int main() {
        int x = 10;
        x = x + 10;
        printf("%s", x);
        return 0;
    }
```

(a) 10
(b) 20
(c) 30
(d) Undefined behavior.

---

18. **What is the purpose of the `enum` keyword?**

(a) To define a set of named integer constants
(b) To declare a variable
(c) To define a macro
(d) To define an array

---

19. **What is the output of the following code?**

```
    #include <stdio.h>
    int main() {
        int x = 10;
        x = x << 1;
        printf("%d", x);
        return 0;
    }
```

(a) 10
(b) 20
(c) 30
(d) Undefined behavior.

---

20. **What is the purpose of the `union` keyword?**

(a) To define a structure.

(b) To define a set of variables that share the same memory location.

(c) To define a macro.

(d) To include a library.

---

21. **What issues exist in the following recursive function, and how can they be corrected? (Select all that apply!)**

```c
#include <stdio.h>

int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}

int main() {
    printf("%d\n", factorial(-5));
    return 0;
}

```

(a) The base case should check for `n <= 0` instead of `n == 0` to prevent infinite recursion.

(b) The function should return an error message or handle negative values before recursion starts.

(c) The function should use iteration instead of recursion to avoid deep recursion for large values.

(d) No changes needed; the function works correctly.

---

22. **What is the purpose of the `bit field` in C?**

(a) To define a memory-efficient structure.

(b) To define a set of variables that share the same memory location.

(c) To define a macro.

(d) To include a library.

---

23. **What is the output of the following code?**

```c
#include <stdio.h>
int main() {
    int x = 10;
    x = x & 1;
    printf("%d", x);
    return 0;
}

```

(a) 10

(b) 1

(c) 0

(d) Undefined behavior.

---

24. **Which of the following statements about `gcc` compilation are true? (Select all that apply!)**

(a) The `-o` option is used to specify the output file name.

(b) The `-Wall` flag enables most common compiler warnings.

(c) The `gcc` command can generate executable files from C source code.

(d) The `-g` flag is used to include debugging information in the compiled output.

---

25. **What is the output of the following code?**

```c
#include <stdio.h>
int main() {
    int x = 10;
    x = x | 1;
    printf("%d", x);
    return 0;
}

```

(a) 10

(b) 11

(c) Compilation error.

(d) Undefined behavior.

Questions 21-24 refer to the following C program, which contains multiple errors. Answer questions 21-24 to identify and correct specific issues in the program:

```c
#include <stdio.h>

int counter(int n) {
    while (n > 0);
        printf("%d ", n);
        n--;
    return n;
}

int main() {
    int x = 5;
    counter(x);
```

```
13
14          int i = 0;
15          while (i < 3) {
16              static int count = 0;
17              count++;
18              printf("Count: %d\n", count);
19          }
20
21          char str[4] = "Test";
22          printf("%s\n", str);
23
24          return 0;
25      }
26
```

26. **Why does the function `counter()` (lines 3-8) result in an infinite loop, and how should it be corrected?**

    (a) Remove the semicolon after the `while` condition

    (b) In line 4, change statement '`while (n > 0);`' to '`do { ... } while (n > 0);`'

    (c) In line 4, change statement '`while (n > 0);`' to '`while (n-- > 0);`'

    (d) No changes needed; the function executes fine

---

27. **What is the logical issue with the `while` loop in `main()` (lines 15-19), and how can it be fixed?**

    (a) The loop is missing an increment statement for the variable `i`

    (b) In line 16, move the declaration `static int count = 0;` outside the loop.

    (c) `static int count=0;` must be `int count=0;`

    (d) No changes needed; the loop executes fine.

---

28. **What is the issue with the string initialization (line 21), and how should it be corrected? (Select all that apply!)**

```
1       char str[4] = "Test";
2
```

    (a) Increase the array size to `char str[5];` to accommodate the null terminator

    (b) Change `char str[4] = "Test";` to `char str[5] = {'T', 'e', 's', 't', '\0'};`

    (c) Use `char str[] = "Test";` instead, letting the compiler determine the correct size.

    (d) No changes needed; the program runs fine.

---

29. **The function `counter()` is intended to print numbers in descending order, but its return value may not work as expected. What issue exists with the return statement, and how can it be corrected?**
Expected Correct Output (After Fixes):

```
1       5 4 3 2 1
2
```

(Select all that apply!)

    (a) The function doesn't return any value; changing its return type to `void` would be better.

    (b) The return statement should be inside the while loop to exit the function when `n` reaches 0.

    (c) The function should decrement `n` before printing it to ensure correct output.

    (d) No changes needed; the function behaves as expected.

---

30. **Which gdb command is specifically used to set a breakpoint at the start of the `main` function, ensuring execution pauses when `main` is reached?**

    (a) `break main`

    (b) `run main`

    (c) `start`

    (d) `None`
