

112-2 Introduction to Programming (EE & CE)

Midterm Exam

Date: April 17, 2024

Time: 16:20 ~ 17:35 (75 minutes)

(Max 100pt / Total 100pt)

Note: This is a **Close Book** examination. All the problems are described in English. You can write your answers in English or Chinese. In addition, **no portable device** is allowed during the exam.

A. [30%, 3% per question] True/False Questions (Please answer T or F.)

- (1) Statements in a high-level language are converted to statements in machine language by a loader.
- (2) Because the following statement is a valid call to function `wow`, `wow` must be a void function.

```
num = wow(x, y);
```
- (3) The C language is case sensitive.
- (4) If the value of `x` is 635, the statement

```
printf("%4d", x);
```


will display four blanks followed by 635.
- (5) The loop repetition condition of a `while` or `for` statement can be false before the loop begins to execute.
- (6) All values stored in memory are represented as binary strings, patterns of zeros and ones.
- (7) Pseudocode is a special form of machine language produced by the C compiler.
- (8) The expression `'e' < 'F'` is true in ASCII.
- (9) If `b` is an array with type `int` elements and the value of `b[4]` is 3, then the statement

```
printf("%d\n", b[b[4] - 1]);
```


displays one less than the value of `b[3]`.
- (10) If `name` is a string variable whose value is "Elizabeth", the function call

```
strcpy(target, &name[5]);
```


assigns the string "beth" to `target`.

B. [30%, 3% per question] Single Choice Questions

- (11) Which statement is false?
 - (a) When a pointer is defined but not initialized, it will automatically point to NULL.
 - (b) B function's prototype declaration can be placed inside function A.
 - (c) To encourage software reusability, it is recommended to pass the size of the array along with the array itself.
 - (d) Using global variables may violate the principle of least privilege, but can be necessary in certain scenarios.
- (12) Which of the following statements is true for the requirements for argument list correspondence?
 - (a) provide the required number of arguments
 - (b) make sure the order of arguments is correct
 - (c) make sure that each function argument is the correct type
 - (d) all of the above
 - (e) none of the above

(13) What is the complement of the following expression?

$n \parallel a \leq b \ \&\& \ c \neq 100$

- (a) $!n \parallel a > b \parallel c == 100$
- (b) $!(n \ \&\& \ (a > b \parallel c == 100))$
- (c) $!n \ \&\& \ (a > b \parallel c == 100)$
- (d) $!(n \parallel (a > b \parallel c == 100))$
- (e) none of the above

(14) What value is returned by function result?

```
int result(const int a[], int n)
{
    int i, r;
    r = 0;
    for (i = 1; i < n; ++i)
        if (a[i] > a[r])
            r = i;
    return (r);
}
```

- (a) The subscript of the largest of the first n elements of array a.
- (b) The value of the largest of the first n elements of array a.
- (c) The subscript of the smallest of the first n elements of array a.
- (d) The value of the smallest of the first n elements of array a.
- (e) The subscript of the last element greater than its predecessor within the first n elements of array a.

(15) Which character marks the end of a string?

- (a) void
- (b) *
- (c) \0
- (d) \!
- (e) none of the above

(16) Which one of the conditions that follow will be false (value of 0) after execution of the program segment below?

```
int v[5] = {0, 0, 0, 0, 1};
int k, j;
for (j = 3; j >= 0; --j)
    for (k = j; k < 4; ++k)
        v[k] += v[k + 1];
```

- (a) $v[0] == v[4]$
- (b) $v[1] == v[3]$
- (c) $v[0] < v[1]$
- (d) $v[1] < v[2]$
- (e) $v[2] < v[3]$

- (17) To test whether a character is one of '0', '1', '2', '3', '4', '5', '6', '7', '8', or '9', use the _____ standard library function.
- (a) isnumber
 - (b) isnotalpha
 - (c) isnumeric
 - (d) isdigit
 - (e) all of the above
- (18) Advantages of typedef do not include
- (a) Increasing the efficiency of accessing struct member variables
 - (b) Making type names shorter.
 - (c) Making programs more readable.
 - (d) Making programs more portable by allowing data types to be easily changed to meet system specifications.
- (19) In a flowchart of an algorithm, what is the shape of the decision symbol?
- (a) circle
 - (b) rectangle
 - (c) diamond
 - (d) rounded rectangle
- (20) Which of the following is not a valid escape sequence?
- (a) \n
 - (b) \\
 - (c) \~
 - (d) \"

C. [40%] Short Answer Questions

(21) Consider the following program and the expected output, answer the following questions:

- (a) [4%] Please filled the blank (a).

Program:	<pre> #include <stdio.h> #include <stdlib.h> int main(void) { // loop 20 times for (unsigned int i = 1; i <= 20; ++i) { // pick random number from 1 to 6 and output it printf("%10d", _____ (a) _____); // if counter is divisible by 5, begin new line of output if (i % 5 == 0) { puts(""); } } } </pre>
----------	--

Output:	6	6	5	5	6
	5	1	1	5	3
	6	6	2	4	2
	6	2	3	4	1

- (b) [4%] Which function is used to specify a seed value for a random number function to randomize number generation?
- (22) [10%] What is the compilation process of C programs? Please write/draw a flow chart of the compilation process.
- (23) [12%] The following is the bubble sorting function, which is a programmatic method for sorting the contents of a sequence from small to large. Please fill in the program block in the dotted box with the most suitable C program statements.

```
#include <stdio.h>
#define SIZE 10
// function main begins program execution
int main(void)
{
    // initialize a
    int a[SIZE] = {2, 6, 4, 8, 10, 12, 89, 68, 45, 37};
    puts("Data items in original order");

    // output original array
    for (size_t i = 0; i < SIZE; ++i) {
        printf("%4d", a[i]);
    }

    // bubble sort
    // loop to control number of passes
    for (unsigned int pass = 1; pass < SIZE; ++pass) {

        // loop to control number of comparisons per pass
        for (size_t i = 0; i < SIZE - 1; ++i) {

            // compare adjacent elements and swap them if first
            // element is greater than second element
            if (a[i] > a[i+1]) {
                .....;
                .....;
                .....;
            }

        }

    }

    puts("\nData items in ascending order");

    // output sorted array
    for (size_t i = 0; i < SIZE; ++i) {
        printf("%4d", a[i]);
    }
    puts("");
}
```

(24) Mathematically, Fibonacci numbers are defined recursively as follow:

$$F_n = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ F_{n-1} + F_{n-2}, & \text{if } n \geq 2 \end{cases}$$

Please fill in the program block in the dotted box with the most suitable C program statements.

```
#include <stdio.h>
unsigned long long int fibonacci(unsigned int n); // function prototype
int main(void)
{
    unsigned int number; // number input by user
    // obtain integer from user
    printf("%s", "Enter an integer: ");
    scanf("%u", &number);

    // calculate fibonacci value for number input by user
    unsigned long long int result = fibonacci(number);
    // display result
    printf("Fibonacci(%u) = %llu\n", number, result);
}
// Recursive definition of function fibonacci
unsigned long long int fibonacci(unsigned int n)
{
    -----
    if (.....) {
        return n;
    }

    else {
        return .....
    }
    -----
}
```

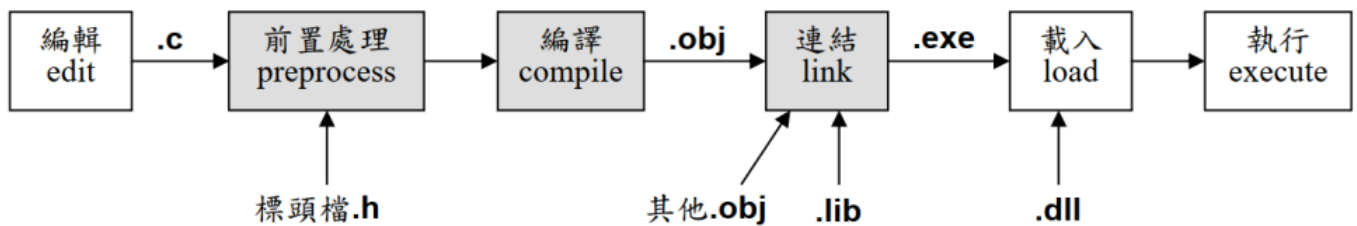
Midterm Answer Sheet

Class : _____ ID : _____ Name : _____

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
F	F	T	F	T	T	F	F	F	T
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
a	d	c	a	c	e	d	a	c	c

(21a)	<code>1 + rand() % 6</code>
(21b)	<code>void srand(unsigned int seed)</code> 有寫 <code>srand</code> 就給分

(22)



(23)

```

if (a[i] > a[i+1]) {
    int temp= a[i];
    a[i] = a[i+1];
    a[i+1] = temp;
}

```

(24)

```

if (n==0 || n==1) {
    return n;
}
else{
    return fibonacci(n-1) + fibonacci(n-2);
}

```

~Good luck with your exam~