Computer Programming - Midterm Mock Exam - Question Sheet

考試時間 Test time 15:50 pm to 18:10 pm; 2.5 hours. 以教室後方時鐘為準 Using the clock at the back of the classroom as official time. 在答案卷上作答 Answer on the answer sheet. 每題一分 One point per problem. 全對才給分 No partial credit. 不倒扣 No penalty for wrong answers. 禁止外部資源（比照學測、分科） No external resources, like TOEFL.

1 Program Meta

```
8888888888888888888888888888888888888888
8888888888888888888889888888888888888888
8888888888888888888888888888888888888888
8888888888888888888888888988888888888888
8888888888888888888888888888888888888888
8888888888888888888888888888888888888888
8888888888888888888888888888888888888888
8888888888888888888888888888888888888888
8838888888888888888888888888888888888888
8888888888888888888888888888888888888888
```

[1] Find a hidden 9.

[2] Find another hidden 9.

[3] Find the other hidden number.

[4] C is invented in (A) 1972 (B) 1982 (C) 1992 (D) 2002

[5] C++ is invented in (E) 1953 (F) 1963 (G) 1973 (H) 1983

```
#include <iostream>
using namespace std;
int main() {
    cout << "hello"
         << "ntu";
}
```

[6] The code to the right outputs (I) `he11ontu` (J) `hell0ntu` (K) `hellontu`

[7] What if we `cout << "hello" "ntu";`? Select two correct answers. (L) It is a syntax error. (M) It outputs `hellontu`. (N) It outputs `hello ntu`. (O) String literals are concatenated.

[8] The purpose of using namespace `std` is to (P) follow the C++26 standard for variable names (Q) access the objects without repeating the prefix (R) follow the latest C++ standard for naming conventions

[9] For a beginner who just finished the `hello world` program, a good next step is to write (S) a `hello world` program in python and cython (T) a program that takes two strings and prints them (U) a program that can draw Hilbert curve using turtle

[10] `#include <iostream>` includes (V) the entire C++ standard library (W) the implementation of `cout` and `cin` (X) a header file that help you read from keyboard (Y) the interface to file-in and file-out operations

```
int main (int argc, char *argv[]) {
    return 0;
}
```

[11] Which two are correct about `main`? (Z) It should always return `0`. (A) It is the only function allowed in a C++ program. (B) It is special in that you cannot call it recursively. (C) For `g++-14`, `argv` may contain things like `-o` and `-Wall`.

```
g++-14 -Ofast -Wall option_pricing.c -o op
op <<< "100 0.05 0.2 1 100"
```

[12] Which one is incorrect regarding these terminal commands? (D) `-o op` specifies the output file name. (E) `g++-14` refers to the compiler developed by GNU. (F) `-Wall` turns on some warnings that sometimes helps finding bugs. (G) `option_pricing.c` is a C file, so `g++` cannot possibly compile it. (H) `-Ofast` is an optimization flag; it tries to make the program faster.

[13] Same as above, choose all incorrect statements. (I) <<< suggests that op will `cout` something. (J) To post-process the output of op, we can use `|` (K) If something goes wrong, use `code option_pricing.c` to edit. (L) `100 0.05 0.2 1 100` will become `argv` in the `main` function (M) If <<< is replaced by <<, it suggests that op will `cin` something.

[14] What command line tools outputs `00000000 cf fa ed fe 07 00 00 01 03 00 00 00 02 00 00 00 |................|` (N) cat (O) vim (P) touch (Q) hexdump

[15] What does the command line tool `grep` do? (R) It searches for matching text. (S) It visualizes data with ASCII art. (T) It compiles `Go representation` code. (U) It grabs and prints the environment variables.

[16] What does command line command `cd` stands for? (V) clear display (W) change directory (X) create dictionary (Y) compile definition

2 Variables and Types

[17] A shadowed variable (Z) is not properly initialized (A) is highlighted in the text editor (B) shares the same name with another (C) is never highlighted in the text editor (D) is an implicit variable created by the compiler

[18] Which are valid variable names? (E) `variablename` (F) `VariableName` (G) `*variablename` (H) `_variablename` (I) `0variablename`

[19] What is `date` after `float date = 2025 / 10 / 15;`? ____

[20] What is the range of 32-bit signed integer? (J) 0 to 2^{31} (K) 0 to 2^{32} (L) 0 to $2^{32} - 1$ (M) -2^{31} to 2^{31} (N) -2^{31} to $2^{31} - 1$

[21] What is the largest `n` such that `signed char fibonacci(int n)` does not overflow, with the initial condition that `fibonacci(2)` is 2. (O) 9 (P) 10 (Q) 11 (R) 12

[22] Regarding the lengths of variable names, what should we, the programmer, keep in mind? (S) Longer names makes the compiler slower. (T) Longer names makes the program clearer. (U) Shorter names makes the compiler safer. (V) Shorter names makes the program faster.

[23] `cout << int('c') << int('b');` outputs 9998.
Accordingly, `cout << int('x') << int('y') << int('z');` outputs ____________

[24] `cout << int('X') << int('Y') << int('Z');` outputs ________

[25] What is camel case? (W) A `switch` statement without `break`. (X) The default branch of an `if` statement. (Y) Capitalizing the first letter of each word. (Z) A functional wrapper protecting camel functions.

[26] The binary representation of `int16_t a = 1111` is ____________________

[27] The binary representation of `int16_t a = -12` is ____________________

[28] How many bytes of memory does `int16_t A[100000];` need? (A) 100000 (B) 200000 (C) 400000 (D) 800000 (E) 1600000

[29] `float` is (F) always 16-bits (G) sometimes 16-bits (H) at least 32-bits (I) at most 32-bits

[30] To the right are some floating-point numbers and their binary representations. What standard is that? (J) IEEE 754 (K) IEEE 768 (L) IEEE 3.14 (M) IEEE 802.11

[31] How much precision is that? (N) half-precision (O) single-precision (P) double-precision (Q) quadruple-precision

[32] What is the binary representation of 64? ____

```
00111111100000000000000000000000 = 1
01000000000000000000000000000000 = 2
01000000010000000000000000000000 = 3
01000000100000000000000000000000 = 4
01000000101000000000000000000000 = 5
01000000110000000000000000000000 = 6
01000000111000000000000000000000 = 7
01000001000000000000000000000000 = 8
01000001000100000000000000000000 = 9
01000001001000000000000000000000 = 10
```

[33] What is the binary representation of `-0.0625`? ____

[34] What is the binary representation of `3.5`? ____

[35] In the range of `int`, `1010 + 1020 + 1030 + 1040` is (R) `5000` (S) `6174` (T) `4100` (U) overflow

[36] Which of the following is the most common way to define constant `PI` to the precision of 32-bit `float`? (V) `float PI = 3.14;` (W) `float PI = 3.1415927;` (X) `float PI = 3.1415926535 8979323846;` (Y) `float PI = 3. 1415926535 8979323846 2643383279 5028841971 6939937510 5820974944 5923078164 0628620899 8628034825 3421170679;`

3 Operators and Arithmetic

[37] What is `3 << 3`? __

[38] What is `'x' ^ 'X'`? (Z) 0 (A) 1 (B) 2 (C) 4 (D) 8 (E) 16 (F) 32 (G) 64 (H) 128 (I) 256

[39] How to check if `n` is a multiple of 17? (J) `n / 17 == 0` (K) `n % 17 == 0` (L) `n // 17 == 0` (M) `n %% 17 == 0`

[40] What happens after `b += a;`? (N) a increases by b (O) b increases by a (P) a becomes positive b (Q) b becomes positive a

[41] If `int a = 3, b = 2;`, select all that compile. (R) `cout << a + b;` (S) `cout << a + + b;` (T) `cout << a ++ + b;` (U) `cout << a + ++ b;` (V) `cout << a ++ + + b;` (W) `cout << a + + ++ b;` (X) `cout << a ++ + ++ b;`

[42] What is the overall output of the choices above (after removing choices that do not compile)? __________

[43] Finish this function that checks whether `n` is a perfect square. You can assume that `n` is between `0` and `100`.

____

```
bool isSquare (int n) {
    // your code here
    // your code here
    return false;
}
```

[44] Use this function `int max3(int a, int b, int c);` to express the medium of `int x, y, z;`. (No sorting because it is not an array.)

____

[45] Body mass index is the quotient of weight in kilograms and height in meters squared. Complete the code. ______________________________

```
cout << "Height centimeters?";
cin >> height;
cout << "Weight in kilograms?";
cin >> weight;
cout << "BMI is " << /*your code here*/;
```

[46] Select all that are true. (Y) `pn -> n` makes `pn` a pointer to `n`. (Z) `A => B` asserts that `B` is a derived class of `A`. (A) `a - - b` is equal to `a + + b` for integers `a` and `b`. (B) `(a * b) * c` must equal to `a * (b * c)` for `float a, b, c` if there is no overflowing or underflowing.

[47] What is the most readable yet correct way to assign 4 percent of 25 to neo? (C) `float neo = 4% * 25;` (D) `float neo = 25% * 4;` (E) `float neo = 0.05 * 25;` (F) `float neo = 4 / 100. * 25;`

4 Flow Control

[48] To the right is the Pascale triangle. For a given input `n`, output the first `n` rows of the Pascal triangle in the same format. Except that all numbers are `% 3;` ___________________

```
1 1 1 1
1 2 3
1 3
1
```

[49] Select true statements. (G) `for(;;)` is an infinite loop. (H) `for(ture)` is an infinite loop. (I) `while(1)` is an infinite loop. (J) `while(2)` cannot be stopped by `break`.

[50] Select true statements. (K) `break` will break the inner-most loop. (L) `continue` will restart the

outer-most loop. (M) `fbreak` will break the inner-most for-loop. (N) `wbreak` will break the outer-most while-loop.

```
float f = 0;
float dfdt = 1;
float dt = 1e-3;
for (float t = 0; t < 10; t += dt) {
    float ddfddt = -f;
    dfdt += ddfddt * dt;
    f += dfdt * dt;
}
```

[51] Some numerical ODE code is presented to the right. What is the equation being solved? (O) $f' = f$ (P) $f'' = f$ (Q) $f' = -f$ (R) $f'' = -f$

[52] Select the impliedinitial conditions. (S) $f(1) = 0$ (T) $f(0) = 0$ (U) $f'(0) = 0$ (V) $f'(0) = 1$

[53] Select all that apply. (W) In industry, there are better methods. (X) `1e3` and `0x1e3` are of the same type. (Y) The loop runs for about 100000 iterations. (Z) `t` is casted into an `int` before comparing to `10`.

```
int S = 0;
for (int i = 0; i < 100000000; i++) {
   if (rand() < 0.003) { S++; }
}
```

[54] After the loop to the right, with high probability, `S` is closest to (A) 300 (B) 3000 (C) 30000 (D) 300000

[55] Recall that modern CPUs have frequency ranging from 1 to 5 GHz. How long does it take to run the loop in the previous question? (Assuming no flags like `-Ofast`) (E) about 1–10 microseconds (F) about 1–10 seconds (G) about 1–10 days (H) about 1–10 years

5 Arrays

[56] What does `int A[10][20] = {};` do? Select all that apply. (I) It sets all 200 entries to zero. (J) It is faster than `int A[10][20];` (K) It allocates some space for 200 integers. (L) It needs some space for 10 pointers to integers.

[57] Declared as `int A[100];` the type of `A` is best described as (M) an array (N) an integer array (O) an array of length 100 (P) an array of 100 integers

[58] Which one sums all elements of `A`? (Q)

```
for(i = 1; i <= 100; i++)
    sum += A[i];
```

(R)

```
for(i = 99; i >= 0; --i)
    sum += A[i];
```

(S)

```
for(i = 1; i < 100; ++i)
    sum =+ A[i];
```

```
int A[10] = {/*hidden*/};
int max0 = 0;
for (int j = 0; j < 10; j++) {
   max0 = max0 < A[j] ? max0 : A[j];
}
```

[59] When the loop ends, `max0` is (T) `0` if all `A[j]` are positive (U) the maximum of an integer array `A` (V) `inf` if `A` contains negative numbers (W) undeclared if the array `A` is not initialized

[60] When the loop ends, `j` is (X) 0 (Y) 9 (Z) 10 (A) not usable, as `j` is local to the loop

[61] Joseph wants to solve heat equation $\frac{du}{dt} = \frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}$ using finite element method. What seems to be wrong with his code?

```
void EvolveOneSecond(float Temperature[100][100], float conductivity) {
    float Temporary[100][100];
    for (int x = 0; x < 100; x++) {
        for (int y = 0; y < 100; y++) {
            float ThisT = Temperature[x][y];
            float SumDiff = 0;
            SumDiff += Temperature[x-1][y] - ThisT;
            SumDiff += Temperature[x+1][y] - ThisT;
            SumDiff += Temperature[x][y-1] - ThisT;
            SumDiff += Temperature[x][y+1] - ThisT;
            Temporary[x][y] = SumDiff * conductivity;
        }
    }
    for (int x = 0; x < 100; x++) {
        for (int y = 0; y < 100; y++) {
            Temperature[x][y] = Temporary[x][y];
        }
    }
}
```

[62] There are two DNA sequences `char A[200], B[200];`. Knowing that they came from a very old, therefore noisy, machine, how do I check if the second half of `A` matches the first half of `B`? (B) find the longest common substring of `A` and `B` (C) find the shortest common supersequence of `A` and `B` (D) find the longest palindromic substring of `A[100..200] + B[0..100]` (E) compute the Hamming/Levenshtein distance between `A[100..200]` and `B[0..100]`

6 Functions

[63] Select the three scenarios where your function needs to take another function as an argument. (F) To sort an array (G) To reverse an array (H) To time a function (I) To remove nans from an array (J) To define a recursive function (K) To cache the values of a function

[64] Using a, b, c, d, x, `add( , )`, and `mul( , )`, instead of `+` and `*`, How to express cubic polynomial $ax^3 + bx^2 + cx + d$?

```
float f(float x) { return 10 * x + 1; }
float f(int x) { return 10 * x + 2; }
int g(float x) { return 10 * x + 3; }
int g(int x) { return 10 * x + 4; }
```

[65] According to the function definitions, what is `f(5)`? __

[66] What is `g(1)`? __

[67] What is `f(g(1))`? ___

[68] What is `f(f(g(g(f(1)))))`? _______

```
#include <iostream>
#include <random>
using namespace std;
int main () {
    mt19937 rand(20251015), rane(20251015), ranf(20251015);
    cout << rand() << " " << rand() << " " << rand() << endl;
    cout << rane() % 2 << endl;
    cout << ranf() % 3 << endl;
    cout << rane() % 2 << rane() % 2 << endl;
    cout << ranf() % 3 << ranf() % 3 << endl;
}
```

[69] The first line of the output is `2357136044 2546248239 3071714933`. What is the second line of output? _

[70] What is the third line of output? _

[71] What is the fourth line of output? __

[72] What is the fifth line of output? __

7 Pointers and Memory

[73] `int *a[10];` declares a length-10 array of pointers to integers. How to declare a pointer to an array of 10 integers? ___________

[74] What is the correct way to declare two pointers, both pointing to integers? (L) `int* a, b;` `(M) int *a, *b;` `(N) &int a, b;` `(O) int &a, &b;`

```
int a = 10;
int *p = &a;
*p += 20;
cout << a;
```

`[75] The code to the right outputs (P) 10`(Q) `20`(R) `200`

[76] Why is `malloc(100 * sizeof(int))` preferred over `malloc(400)`? (S) Because it is faster to avoid computer `sizeof(int)`. (T) Because `int` may be 4 bytes on some machines but 8 bytes on others. (U) Because `sizeof(int)` is 8, so `400` is half of what we actually need. (V) Because `sizeof(int)` is 2, so `400` is twice of what is actually needed.

```
int a = 1, b = 2, c = 3;
int *p = &a, *q = &b, *r = &c;
int **x = &p, **y = &q, **z = &r;
cout << **x << **y << **z << endl;
swap(x, y);
cout << **x << **y << **z << endl;
swap(a, c); swap(q, r);
cout << **x << **y << **z << endl;
swap(*x, *z);
cout << **x << **y << **z << endl;
swap(*p, *q); swap(**y, **z);
cout << **x << **y << **z << endl;
```

[77] Look at the code to the right, what is the first line of output? ___

[78] What is the second line of output? ___

[79] What is the third line of output? ___

[80] What is the fourth line of output? ___

[81] What is the fourth line of output? ___

[82] `1*a**p***x` is _

8 Structures and Classes

```
struct Circle { float x, y, r; };
struct Circle Scale (struct Circle c, float s) {
    // your code here
    return big;
}
```

[83] Finish the function `Scale` such that `Scale(c, 2)` shares the same center with `c` but has double the radius.

[84] `RotateCCW` keeps the center of a rectangle and rotate it by 90 degrees CCW.

```
struct Rectangle { float x1, y1, x2, y2; };
struct Rectangle Scale
(struct Rectangle c, float h, float v) {
    // your code here
    return big;
}
```

[85] Keep the center, scale the rectuangle using `h` as horizontal multiplier and `v` as vertical multiplier.

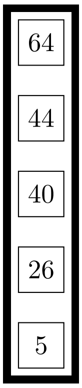
```
structure Node {
    double a;
    Node *b;
    Node *c;
}
```

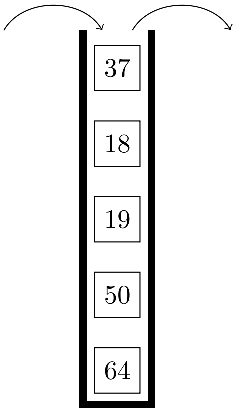
[86] Reorder the following lines to make sense. (W) `tail->next = new Node; tail = tail->next; tail->data = count++; tail->next = NULL;` (X) `Node *head = new Node; head->data = count++; head->next = NULL;` (Y) `struct Node { int data; Node *next; };` (Z) `Node *tail = head;` (A) `int count = 0;`

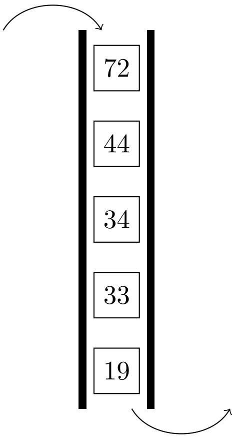
```
string Cmplx2Strng (struct Cmplx z) {
    string answer = to_string(z.a);
    answer += " + ";
    answer += to_string(z.b);
    answer += "i";
    return answer;
}
```

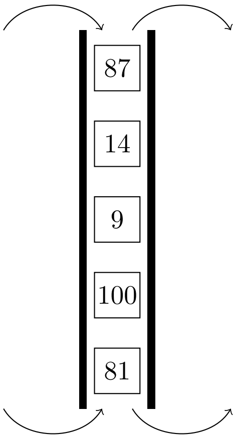
[87] Recall that $i = \sqrt{-1}$. Assume that I use the code to the right to print complex numbers. Guess the structure of `Cmplx`.

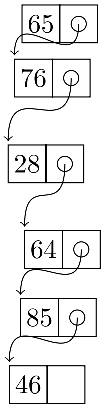
[88] When a stack is empty, what happens if we try to pop? (B) The stack will overflow. (C) The stack turns into a queue. (D) The stack enters an infinite loop. (E) It depends on the implementation.

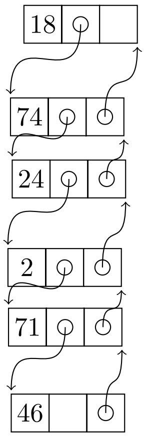
(F)


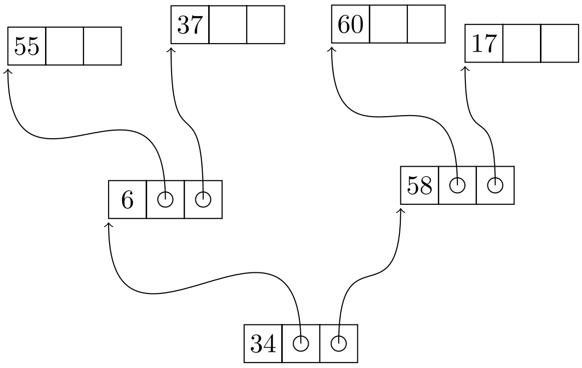
(G)


(H)


(I)


(J)


(K)


(L)


[89] Which drawing represents a fixed-length array?

[90] Which drawing represents a stack (LIFO)?

[91] Which drawing represents a queue (FIFO)?

[92] Which drawing represents a double-ended queue?

[93] Which drawing represents a (singly) linked list?

[94] Which drawing represents a doubly linked list?

[95] Which drawing represents a binary tree?

9 Algorithm

[96] Consider the program to the right. You want to make it as hard as possible to a beginner, but you also want to ensure that you yourself can always win. What should you set `attempts` to? (M) 1 (N) 2 (O) 3 (P) 4 (Q) 5 (R) 6 (S) 7 (T) 8 (U) 9 (V) 10 (W) 11 (X) 12 (Y) 13 (Z) 14 (A) 15 (B) 16 (C) 17 (D) 18 (E) 19 (F) 20

```
int attempts = ???;
int guess = 0;
int target = rand() % 100;
while (attempt > 0) {
    attempt -= 1;
    cin >> guess;
    if (guess > target) { cout << "Too high"; }
    else if (guess < target) { cout << "Too low"; }
    else { cout << "Win"; return; }
}
cout << "Lose: no more attempts";
```

[97] What is the purpose of the function to the right? (G) To sort an array (H) To rotate an array (I) To revert an array (J) To shuffle an array (K) To summarize an array

```
void BreadthOfTheField(int A[], int a, int b) {
    while (a < b) {
        int t = A[a]; A[a] = A[b - 1]; A[b - 1] = t;
        BreadthOfTheField(A, a + 1, b - 1);
        break;
    }
}
```

[98] Consider another function as show to the right. This is my first implementation that uses a second array B. I want to achieve the same effect, but without using extra memory (i.e., the second array B). So, instead, I call `BreadthOfTheField` three times. With what parameters should I call it? ______________________________

```
void TrickOfTheKitchen(int A[100], int c) {
    int B[100];
    for (int i = 0; i < 100; i++) {
        B[i] = A[(i + c) % 100];
    }
    for (int i = 0; i < 100; i++) {
        A[i] = B[i];
    }
}
```

Computer Programming - Midterm Mock Exam - Answer Sheet

Name (zh or en) Hsin-Po Wang Student ID B00201054

[1] [2] [3] [4] A [5] H [6] K [7] M O [8] Q [9] T [10] X [11] B
C [12] G [13] I L [14] Q [15] R [16] W [17] B [18] E F H [19]
13 [20] N [21] P [22] T [23] 120121122 [24] 888990 [25] Y
[26] 0000010001010111 [27] 1111111111110011 [28] B [29]
H [30] J [31] O [32] 0 10000101 0000000000000000000
[33] 1 01111011 00000000000000000000000 [34]
0 10000000 11000000000000000000000 [35]
T [36] W [37] 24 [38] F [39] K [40] O [41] R S T U V W X [42]
55577910 [43] if (n == int(sqrt(n)) * int(sqrt(n))) ret
[44] x + y + z - max3(x, y, z) + max3(-x, -y, -z); [45]
weight * 10000. / height / height [46] A [47] F [48]
for (...) {for (...) {...}} [49] G I [50] K [51] R [52] T V [53]
W [54] A [55] F [56] I K [57] P [58] R [59] T [60] A [61] [62] E [63]
F H K [64] [65] 52 [66] 54 [67] 542 [68] 523421 [69] 1 [70]
2 [71] 01 [72] 00 [73] int (*a)[10]; [74] M [75] R [76] T [77]
123 [78] 213 [79] 132 [80] 231 [81] 312 [82] 3 [83] [84] [85] [86]
W X Y Z A [87] [88] E [89] [90] [91] [92] [93] [94] [95] [96] S [97] I [98]
(A, 0, c); (A, c, 100); (A, 0, 100);