

Midterm Test - AY18/19

Question 1: C/C++ Expressions

There are several parts to this question which are to be answered independently and separately. Each part consists of a fragment of C/C++ code. Write the exact output produced by the code in the answer box. If an error occurs, or it enters an infinite loop, state and explain why. You may show workings outside the answer box in the space beside the code. Partial marks may be awarded for workings if the final answer is wrong. Assume that all appropriate preprocessor directives e.g., `#include <iostream>`, etc. have already been defined.

```
In [1]: #include <iostream>
        using namespace std;
```

A.

```
In [ ]: int f(int x, int y)
        {
            x *= 10;
            y += x;
            return y;
        }

        int main()
        {
            int x = 4, y = 7;
            cout << f(f(y, x), y);
        }
```

Initialize $x = 4$, $y = 7$

0. Call $f(x,y) \rightarrow f(7,4)$

$x = 7 * 10 = 70$

$y = 4 + 70 = 74$

return 74

1. Call $f(x,y) \rightarrow f(74,7)$

$x = 74 * 10 = 740$

$y = 7 + 740 = 747$

return 747

Ans: 747

B.

```
In [ ]: int a = 10;

        if (a > 5)
            cout << "Here ";
        if (a % 5)
            cout << "we ";
        else if (a)
            cout << "go ";
        else
            cout << "again";
```

Initialize $a = 10$

$10 > 5$, so will pass *if*($a > 5$)

Print "Here"

$10 \% 5 = 0$, so will not pass *if*($a \% 5$)

Proceed to *else if* (a)

Print "go"

Ans: Here go

C.

```
In [ ]: for(int i = 10; i > 0; i -= 2)
{
    cout << i << endl;
    if (i % 4 == 0)
        break;
    if (i % 3 == 0)
    {
        i -= 3;
        continue;
    }
    i += 1;
}
```

Initialize i = 10

When i = 10,

Print "10\n"

10 % 4 != 0, will not pass if(i % 4 == 0)

10 % 3 != 0, will not pass if(i % 3 == 0)

Increment i + 1 = 11

continue to loop, decrement i - 2, i = 9

When i = 9,

Print "9\n"

9 % 4 != 0, will not pass if(i % 4 == 0)

9 % 3 = 0, pass if(i % 3 == 0)

Decrement i - 3, i = 6, continue to loop continue to loop, decrement i - 2, i = 4

When i = 4,

Print "4\n"

4 % 4 = 0, will pass if(i % 4 == 0)

Exit loop

Ans:

10

9

4

Question 3: Computing Topics

A. Computer Organisation

i. The major components of a computing device is connected by BUS.

True.

Bus is the bridge between processor & memory. (Lecture03-pg44)

ii. In modern laptops, the SSD (solid state drive) plays the role of the MEM-ORY.

False.

Memory refers to RAM

iii. The registers are storage inside the processor.

True

Programs will be compiled into assembly languages > translated into machine language.(Lecture03-pg39)

iv. The executable compiled by Visual Studio Code, e.g. "task1.exe" is stored in the HARDDISK.

True.

B. If you have a quadcore processor (i.e. 4-core), which of the following is TRUE?

i. A program written for single-core processor can now run up to 4-time faster.

False.

The program has to be written for multi core processor to utilize the computing power.

ii. If each of the core runs at 1 GHz (1 gigahertz), then the quadcore is running at 4 GHz.

False.

1GHz x 4 cores not equals to 4 GHz

iii. There can be up to four program running at the same time on this processor.

True.

iv. It is possible to get 4 times productivity (i.e. more work gets done) in the same amount of time on this processor.

True.

C. Which of the following regarding instruction execution is TRUE?

i. Execution cycle of an instruction may take up to three memory accesses.

True.

Fetch, get operands, store results.

ii. Memory instruction refers only to instruction that moves a value from memory to register.

False.

Also applies to instruction that moves a value from register to memory.

iii. Instruction and data are stored in the same region in memory.

False.

iv. Store-memory concept refers to the idea of storing computation results in the memory.

False.

Instruction and data are stored in memory.

Question 4: Data Representation

A. Which of the following is TRUE regarding the data storage in computer?

i. On a 32-bit platform, each double variable in C/C++ needs 64 bit storage in memory.

True.

For 32/64 bit platforms:

char: 1 byte / 1 byte

short: 2 bytes / 2 bytes

int: 4 bytes / 4 bytes

long: 4 bytes / 8 bytes

long long: 8 bytes / 8 bytes

float: 4 bytes / 4 bytes

double: **8 bytes** / 8 bytes

long double: 16 bytes / 16 bytes

1 bytes = 8 bits, so double needs 64bit.

ii. If a 4-byte word is used to store only non-negative values, then its range is $[0 \dots 2^{32} - 1]$

True.

4 byte = 32 bit, range is 2^{32} , -2,147,483,647 to 2,147,483,648

Sacrifice one bit to indicate whether the number is positive or negative, so the range becomes 2,147,483,647 ($2^{32} - 1$).

iii. Directly comparing floating point values / variables is not safe because of the overflowing problem.

False.

Directly comparing floating point values / variables is not safe because of the inaccuracy.

iv. Computer can store either binary, octal (base-8) or hexadecimal (base-16) values in the actual hardware.

False.

Computer can only store binary on the hardware level.

B. Which of the following number base related statements is TRUE?

i. The decimal number 58_{10} is 10_{58} in base-58.

58_{10}

$= 58 \% 58 = 1 \text{ R } 0$

$= 1 \% 58 = 0 \text{ R } 1$

$= 10_{58}, \text{ True}$

ii. The decimal number 58_{10} is 110010_2 in base-2.

58_{10}

$= 58 \% 2 = 29 \text{ R } 0$

$= 29 \% 2 = 14 \text{ R } 1$

$= 14 \% 2 = 7 \text{ R } 0$

$= 7 \% 2 = 3 \text{ R } 1$

$= 3 \% 2 = 1 \text{ R } 1$

$= 1 \% 2 = 0 \text{ R } 1$

$= 111010_2, \text{ False}$

iii. The number 101111001_2 is an odd number.

Yes, $1 * 2^0$ will get an odd number, hence 101111001_2 is an odd number.

iv. The hexadecimal number $B7_{16}$ is 267_8 in base-8.

Breaking down $B7_{16}$

$7_{16} = 7_8$

$B_{16} = 11 * 16^1 = 176 = 26_8$

Hence, $267_8, \text{ True}$