Loop Statements

Name: Nguyễn Trung Tín
ID: 19127582
Class: 19CLC6

A) Multiple-choice questions (5.0 marks)

A.1. In nested loops, which loop has the largest number of executions?

Answer:

d. This can only be determined when the numbers of executions of all loops are available.

Explain: Most of the time, the answer should be **a. the innermost loop**, but we have a exclusion, the exclusion is when the "innermost loop" loops zero time, and the "outtermost loop" loops more than zero.

Example:

```
for(int i=0; i<3; i++) {
     // do something
     for(int j=0;  false; j++) {
          // do something
     }
}
```

*In the above example, the **outtermost** loop loops 3 times, while the **innermost** loops 0 time. Because the condition (statement 2) is false.*

A.2. Which of the following statements can be used to change the common workflow of a loop?

(There may be multiple valid choices)

Answer:

```
d. All of the above
```

Explain:

The "break" and "exit" are functions that can stop loop, additionally the exit function can also stop the process. So we say that "break" and "exit" functions can change the common workflow of a loop.

Whereas the continue function breaks one iteration, which means it skips lines below till it meet the closing curly brace "}".

A.3. Which type of loop should be used when the number of executions is determined in advance?

Answer:

```
c. for
```

Explain:

We use the for loop because it makes code more readable, so people who read your code can be much more easier to understand.

A.4. At least how many time(s) do the following loops execute? Check appropriate option for

each of the loop or fill in your reasons to the column Others

Loop	0 time	1 time	Infinite	It depends on the condition of the loop	Others
`while`	**x**				
`do…while`		**x**			
`for`	**x**				

A.5. Given that `x` is not affected by the body of the loop, what is the value of `x` after the following code fragment is executed?

```
for (int x=0; x < 100; ++x){
     ...
}
```

Answer:

`e. None of the above.`

Explain:

Because when the for loop reach the "}", "int x" won't available more.

In the above code, x will be undefined.

B. Essay questions (5.0 marks)

B.1. Distinguish between `break` and `continue`. In your opinion, whether using these statements in loops is a good programming habit?

break	continue
Break the loop, means skip all the statement in the nearest loop which contains the break statement. TL;DR: Use to jump out of a loop	Breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.
Break can appear in **switch statement** and **loop statements** (for, while, do - while)	Continue can only appear in **loop statements** (for, while, do-while)
Terminates the nearest block and gets out of the control the switch or loop.	Gets the control to the next iteration of the loop.

Vietnamese: Mình thấy dùng các statement này trong vòng lặp không tốt cũng không xấu. Nếu code bạn viết dễ đọc (readable) thì tốt, còn khó đọc thì không tốt.

B.2. Is it possible to convert a `for` loop to a `do..while` loop? If no, give your reason. If yes, give an example.

Answer: Yes

Example:

For loop

```
for(int i = 0; i < 10; i++) {
    // do something
}
```

While loop

```
int i = 0;
while(i < 10) {
    // do something
    i++;
}
```

B.3. The below code fragment aims to print the numbers from 0 to 9. Point out the error(s) of all types (syntax/semantic/logic), if there is any.

```
Unsigned int i;
     for (i = 9; i >= 0: --i)
          cout << i << endl;
```

Original code	Fixed code	Type of error	Reason (if there is any)
`Unsigned int i;`	`unsigned int i;`	Syntax error	c is case sensitive language, so the keyword must be sensitive too.
`for (i = 9; i >= 0: --i)`	`for (i = 0; i <= 9; i++)`	Syntax error Logic error	Refer to https://www.w3schools.com/cpp/cpp_for_loop.asp Between statement 2 and statement 3 must be ";" The aims is to print the numbers from 0 to 9, not 9 down to 0.

B.4. Correct the syntax error(s) in the following code fragment if there is any. Does the syntax- corrected code fragment work properly? If not, further correct all remaining errors.

```
Int counter = 100;
While (counter < 10){
     cout << counter;
     counter--;
}
```

Syntax Error fix

Original code	Fixed code
`Int counter = 100;`	`int counter = 100;`
While(counter < 10){	while(counter < 10){

Question:

Does the syntax- corrected code fragment work properly?

Answer:

No

Fixed code:

```
int counter = 100;
```

```
while (counter > 10){
     cout << counter;
     counter--;
}
```

B.5. What is the output of the following program? Explain in detail.

```
int main()
{
     int i, j;
     i = j = 2;
     while(--i && j++)
          cout << i << " " <<j;
     return 0;
}
```

1st time running

```
int main()
{
     int i, j;
     i = j = 2;
     while(--i && j++)
         /*
         i and j are now 2.
         In the condition statement above:
              i = 1 (decrease first),
              j = 2 (increase after)
         After the condition statement:
              i = 1, j = 3
         The condition statement return true, then it
         goes to the block code.
         */
         cout << i << " " <<j;  // print "1 3"

     return 0;
}
```

2nd time running

```
int main()
{
     int i, j;
     i = j = 2;
     while(--i && j++)
         /*
         i and j are now 1 and 3
         In the condition statement above:
              i = 0 (decrease first),
              j = 3 (increase after)
         After the condition statement:
              i = 0, j = 4
         The condition statement return false (because
     of the variable i), then it comes through the loop.
     (and then return 0)
         */
```

```
            cout << i << " " <<j;  // print "1 3"

        return 0;
}
```