Assignment 2, Due 2020-09-18,

Estimated time: 30 minutes

Question 1 (Recursive functions): The sequence of Fibonacci numbers $F(n)$ (for all $n \geq 0$):

$$F(n) := \begin{cases} 0, & \text{if } n = 0, \\ 1, & \text{if } n = 1, \\ F(n-2) + F(n-1), & \text{if } n \geq 2 \end{cases}$$

We compute the remainders of $F(n)$, when divided by $p = 2000000011$ (just some large prime):

$$G(n) := G(n) \bmod 2000000011, \quad \text{for } n \geq 0. \tag{1}$$

```
1  #include <iostream>
2
3  const int p = 2000000011;
4
5  int G(int n) {
6    switch(n) {
7      case 0: return 0;
8      case 1: return 1;
9      default: return
10       (G(n-2) + G(n-1)) % p;
11   }
12 }
13
14 int main() {
15   using namespace std;
16   int n; cin >> n;
17   cout << "G(" << n << ")="
18        << G(n) << endl;
19 }
```

Please give YES/NO answers:

(A) The integer `p` (Line 3) should be defined in some class or function; cannot have a variable without a scope.

(B) The `switch` statement should have `break` after every case (Lines 7,8).

(C) The '`using namespace`' has to be before a method, not inside it (Line 15).

(D) The C++ function `G(int n)` uses incorrect algorithm to compute $G(n)$.

(E) The program might be slow for some arguments.

Question 2 (Overloading functions):

```
1  #include <iostream>
2
3  using namespace std;
4  class Square {
5    public:
6    int square(int a) {
7      cout << "square(int)" << endl;
8      return (a*a);
9    }
10   double square(double b) {
11     cout << "square(double)" << endl;
12     return b*b;
13   }
14 };
15
16 int main() {
17   using namespace std;
18   Square ss;
19   cout << ss.square('7') << endl;
20 }
```

Please give YES/NO answers:

(A) Two functions with the same name `square(...)` should have the same return value (either `double` or `int`, but not two at the same time).

(B) Output on Line 19 happens before outputs on Lines 7 and 11.

(C) Char parameter cannot be passed to functions, if their input type is either `int` or `double` (Line 19); you need to write yet another function to square `char` values. E.g.
`double square(char c) { ... }`

(D) Squaring char `'7'` computes $7^2 = 49$, since it is converted to number 7.

Question 3 (Parameters by Value and by Reference):

```
1  #include <iostream>
2
3  using namespace std;
4  void fun(int a, int& b) {
5    a += 10;
6    b += 10;
7    cout << "in fun: (a,b) = (" <<
8      a << "," << b << ")" << endl;
9  }
10
11 int main() {
12   int a = 5;
13   int b = 3;
14   fun(++b,a);
15   cout << "in main: (a,b) = (" <<
16     a << "," << b << ")" << endl;
17 }
```

Write the output produced by this program.

Question 4 (Arrays and Pointers):

```
1  #include <iostream>
2  #include <algorithm>
3
4  int rows = 4, cols = 4;
5
6  using namespace std;
7  void f(int*& a) { a[1] = 101; }
8  void g(int* a) { a[2] = 102; }
9  void h(int*& a) {
10   a = new int[cols];
11   // initialize array with 103:
12   fill_n(a, cols, 103);
13 }
14 void i(int* a) {
15   a = new int[cols];
16   // initialize array with 104:
17   fill_n(a, cols, 104);
18 }
19
20 int main() {
21   int** arr = new int*[rows];
22   for (int i=0; i<rows; i++)
23     arr[i] = new int[cols];
24   f(arr[0]);
25   g(arr[1]);
26   h(arr[2]);
27   i(arr[3]);
28
29   for (int i=0; i<rows; i++) {
30     for (int j=0;j<cols; j++)
31       cout << arr[i][j] << " ";
32     cout << endl;
33   }
34 }
```

What values of `arr` are printed near the end of `main()` function (Lines 29-33)?
Use asterisk `*` to denote those values in the array which may be uninitialized.

Solutions

Question 1. Answer: (E).
The proposed program does compute Fibonacci sequence (and does not contain any errors listed in other answers. But it is extremely inefficient for large arguments, because it calls itself recursively twice.

Question 2. Answer: 3025 (i.e. 55^2).
Char literal `'7'` is converted into `int`. And ASCII code for the symbol `7` is 37_{16}, which is the same as $3 \cdot 16 + 7 = 55$ in decimal.
Booleans and chars are converted into integers, so `square(int)` method is called. If the argument is of type `float`, `double` or `long double`, then `square(double)` method will be called.

Question 3. Answer:

```
in fun: (a,b) = (14,15)
in main: (a,b) = (15,4)
```

In `fun()` we pass the first parameter by value, but the second one – by reference. The function call `fun(++b,a)` increments b (it becomes 4) and then copies it into the function's formal parameter a. It also passes the reference of `a` which becomes the formal parameter b. (The deliberately confusing names a, b just show the fact that the parameter order matters, not their names.)
Inside `fun()` both are incremented by 10, so it outputs $(14, 15)$. After returning from the function a (passed by reference) retains its new value $a = 15$. But b (passed by value) gets its old value, because the copied one was lost. So $b = 4$. So in `main()` we output $(15, 4)$.

Question 4. Answer:

*	101	*	*
*	*	102	*
103	103	103	103
*	*	*	*