2 Code Comprehension

1. `ex1.cpp`

```cpp
#include <iostream>

using namespace std;

int main() {
    int a = 10;
    int b = 11;
    int c = a + b * 3 % 7;
    cout << c << endl;
}
```

What is the output of the given code snippet?

2. `ex2.cpp`

```cpp
#include <iostream>

using namespace std;

int main(int argc, char* argv[]) {
    cout << argv[0] << endl;
    cout << argv[1] << endl;
}
```

This code snippet was compiled using the command `g++ -std=c++11 -ggdb -Wall ex2.cpp -o ex2`. Write two commands to run the produced executable from the command line; one that does not produce an error and one that does produce an error.

3. `ex3.cpp`

```cpp
#include <iostream>

using namespace std;

int main(int argc, char* argv[]) {
    cout << argv[0] << endl;
    cout << argv[1] << endl;
}
```

Modify the given code so that it will not produce a segmentation fault, even if the user runs it with the incorrect number of command line parameters.

4. `ex4.cpp`

```cpp
#include <iostream>

using namespace std;

void Swap1(int x, int y) {
    int tmp = x;
    x = y;
    y = tmp;
}

void Swap2(int &x, int &y) {
      int tmp = x;
      x = y;
      y = tmp;
}

int main(int argc, char **argv) {
      int a = 5, b = 10;

      Swap1(a, b);
      cout << "a: " << a << "; b: " << b << endl;

      a = 5, b = 10;
      Swap2(a, b);
      cout << "a: " << a << "; b: " << b << endl;
      return EXIT_SUCCESS;
}
```

What is the output of the given code?

5. `ex5.cpp`

```cpp
#include <iostream>
#include <cstdlib>
#include <math.h>

using namespace std;

void Mystery(int x) {
    int u = (int)sqrt((double)x);
    bool *a = new bool[x + 1];

    memset(a, 0, sizeof(bool) * (x + 1));      // sets all entries in a to 0

    // loop 1
    for (int m = 2; m <= u; m++) {
        if (!a[m]) {
              // loop 2
            for (int k = m * m; k <= x; k += m) {
                a[k] = true;
            }  // end loop 2
        }
    }  // end loop 1

    // loop 3
    for (int m = 2; m <= x; m++) {
        if (!a[m]) {
            cout << m << " ";
        }
    } // end loop 3

    delete [] a;
}

int main(int argc, char* argv[]) {
    Mystery(atoi(argv[1]));
    return EXIT_SUCCESS;
}
```

(a) What type is `a`?

(b) How many elements does the array hold in relation to `x`?

(c) If `x` is 10, what is the value of `u`?

(d) If `x` is 10, what is the value of `a` after the end of loop 1?

(e) What is the purpose of loop 3?

(f) What does the `Mystery` function do?