Some Practice Problems for the C++ Exam

and Solutions for the Problems

The problems below are *not* intended to teach you how to program in C++. You should not attempt them until you believe you have mastered all the topics on the "Checklist" in the document entitled "Computer Science C++ Exam".

There are 39 problems. The solutions for the problems are given at the end, after the statement of problem 39.

1. What is the exact output of the program below? Indicate a blank space in the output by writing the symbol ␣ . Indicate a blank line in the output by writing blank line .

```
#include <iostream.h>

main()
{
   int n = 4, k = 2;

   cout << ++n << endl;
   cout <<  n  << endl;

   cout << n++ << endl;
   cout <<  n  << endl;

   cout << -n  << endl;
   cout <<  n  << endl;

   cout << --n << endl;
   cout <<  n  << endl;

   cout << n-- << endl;
   cout <<  n  << endl;

   cout << n + k << endl;
   cout <<   n   << endl;
   cout <<   k   << endl;

   cout << n << k << endl;

   cout <<          n << endl;
   cout << "  " << n << endl;

   cout << " n" << endl;
   cout << "\n" << endl;

   cout << " n * n = "; //CAREFUL!
   cout << n * n << endl;
```

```
    cout << 'n' << endl;

    return 0;
}
```

2. What is the output of the program below?

```
#include <iostream.h>

main()
{
   int n = 3;
   while (n >= 0)
   {
      cout << n * n << endl;
      --n;
   }

   cout << n << endl;

   while (n < 4)
      cout << ++n << endl;

   cout << n << endl;

   while (n >= 0)
      cout << (n /= 2) << endl;

   return 0;
}
```

3. What is the output of the program below?

```
#include <iostream.h>

main()
{
   int n;

   cout << (n = 4)  << endl;
   cout << (n == 4) << endl;
   cout << (n > 3)  << endl;
   cout << (n < 4)  << endl;
   cout << (n = 0)  << endl;
   cout << (n == 0) << endl;
   cout << (n > 0)  << endl;
   cout << (n && 4) << endl;
   cout << (n || 4) << endl;
   cout << (!n)     << endl;

   return 0;
}
```

4. What is the output of the following program?

```
#include <iostream.h>

main()
{
   enum color_type {red, orange, yellow, green, blue, violet};

   color_type shirt, pants;

   shirt = red;
   pants = blue;

   cout << shirt << "  " << pants << endl;

   return 0;
}
```

5. What is the output when the following code fragment is executed?

```
int i = 5, j = 6, k = 7, n = 3;
cout << i + j * k - k % n << endl;
cout << i / n << endl;
```

6. What is the output when the following code fragment is executed?

```
int found = 0, count = 5;
if (!found || --count == 0)
   cout << "danger" << endl;
cout << "count = " << count << endl;
```

7. What is the output when the following code fragment is executed?

```
char ch;
char title[] = "Titanic";

ch = title[1];
title[3] = ch;

cout << title << endl;
cout << ch << endl;
```

8. Suppose that the following code fragment is executed.

```
const int LENGTH = 21;
char  message[LENGTH];

cout << "Enter a sentence on the line below." << endl;
cin  >> message;

cout << message << endl;
```

Suppose that in response to the prompt, the interactive user types the following line and presses Enter:

```
Please go away.
```

What will the *output* of the code fragment look like?

9. Suppose that the following code fragment is executed.

```
const int LENGTH = 21;
char  message[LENGTH];

cout << "Enter a sentence on the line below." << endl;
cin.getline(message, LENGTH, '\n');

cout << message << endl;
```

a. Suppose that in response to the prompt, the interactive user types the following line and presses Enter:

```
Please go away.
```

What will the *output* of the code fragment look like?

b. Suppose that in response to the prompt, the interactive user types the following line and presses Enter:

```
Please stop bothering me.
```

What will the *output* of the code fragment look like?

10. Suppose that the following code fragment is executed.

```
const int LENGTH = 21;
char  message[LENGTH];

cout << "Enter a sentence on the line below." << endl;

int i = 0;

do
{
   cin >> message[i];
   ++i;
}
while (i < LENGTH - 1  &&  message[i] != '\n');

message[i] = '\0'; // Terminate string with NUL char.
```