

1. Use some library functions from the lecture 3 handout to write C++ assignment statements to calculate the following formulas:

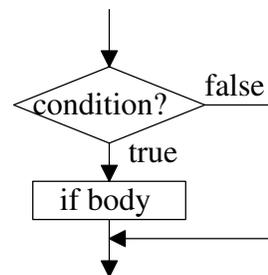
a) The monthly payment on a loan using the formula:  $\text{Payment} = \frac{\text{Rate} \times (1 + \text{Rate})^N}{((1 + \text{Rate})^N - 1)} \times L$ , where Rate is the monthly interest rate (yearly rate / 12), N is the number of payments, and L is the amount of the loan.

b) Generate a random integer in the range 3 to 1000 (inclusive).

2. Recall that an *algorithm* is a sequence of well-defined steps that must be taken in order to perform a task. In English, write an algorithm to exit the building if the fire alarm goes off.

3. An `if` statement allows code to be executed or not based on the result of a comparison. The syntax of an `if` statement is:

```
if (condition) {
    statement1;
    statement2;
    statement3;
} // end if
```



If the condition evaluates to `true`, then the statements of the indented body is executed. If the condition is `false`, then the body is skipped. The statements in the body of the `if` statement are indented.

Typically, the condition involves comparing “stuff” using relational operators (`<`, `>`, `==`, `<=`, `>=`, `!=`). For example, we might want to print “Your grade is A.” if the variable `score` is greater-than or equal to 90.

```
if (score >= 90) {
    cout << "Your grade is A.";
} // end if
```

Complex conditions might involve several comparisons combined using Boolean operators: `!` (not), `||` (or), `&&` (and). For example, we might want to print “Your grade is B.” if the variable `score` is less than 90, but greater than or equal to 80.

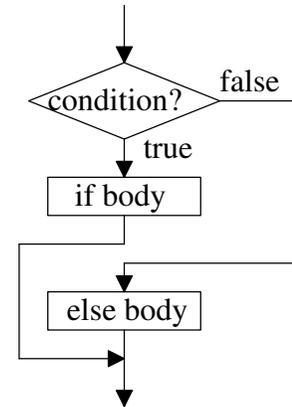
```
if (score < 90 && score >= 80) {
    print "Your grade is B.";
} // end if
```

Using `if` statements, write C++ code to output the appropriate string according to variable `temperature`'s value.

Temperature	String
<code>temperature &lt; 0</code>	“Its bitterly cold!”
<code>0 &lt;= temperature &lt;= 32</code>	“Its freezing outside.”
<code>32 &lt; temperature &lt; 68</code>	“Light jacket weather.”
<code>68 &lt;= temperature</code>	“Its warm outside.”

4. An if/else statement allows a block of code to be executed if the result of a comparison is true; otherwise the “else” block of code will be executed. The syntax of an if/else statement is:

```
if (condition) {
    // only executed if the condition is True
    statementT1;
    statementT2;
    statementT3;
} else {
    // only executed if the condition is False
    statementF1;
    statementF2;
    statementF3;
}
```



Using only nested if/else statements, write C++ code to output the appropriate string according to variable temperature's value.

Temperature	String
temperature < 0	"Its bitterly cold!"
0 <= temperature <= 32	"Its freezing outside."
32 < temperature < 68	"Light jacket weather."
68 <= temperature	"Its warm outside."

5. Using `if/else if` format, write C++ code to output the appropriate string according to variable temperature's value.

Temperature	String
<code>temperature &lt; 0</code>	"Its bitterly cold!"
<code>0 &lt;= temperature &lt;= 32</code>	"Its freezing outside."
<code>32 &lt; temperature &lt; 68</code>	"Light jacket weather."
<code>68 &lt;= temperature</code>	"Its warm outside."

6. Draw a flow-chart for the above `if/else if` formatted code.

7. Correct the syntax and logic errors in the following code:

```
if x < 0 {
    cout << "x = " << x << endl;
    cout << "x is negative." << endl;
} else if (x = 0)
    cout << "x is zero." << endl;
} else if (0 <= x < 10) {
    cout << "x = " << x << endl;
    cout << "x is a small positive." << endl;
} else
    cout << "x = " << x << endl;
    cout << "x is a large positive." << endl;

cout << "Bye!" << endl;
```