# Errata for
# C++ and Algorithmic Thinking
# for the Complete Beginner
## Second Edition

## 14.6 Review Exercises

3. Write a C++ program that prompts the user to enter his or her name and then creates a secret password consisting of three letters (in lowercase) randomly picked up from his or her name, and a random four-digit number. For example, if the user enters "Vassilis Bouras" a secret password can probably be one of "sar1359" or "vbs7281" or "bor1459". Space characters are not allowed in the secret password.

### *Exercise 30.6-2 Rice on a Chessboard*

```
#include <iostream>
using namespace std;
int main() {
  int i;
  unsigned long long grains, total;
  double weight;

  grains = 1;
  total = 1;
  for (i = 2; i <= 64; i++) {
    grains = 2 * grains;
    total = total + grains;
  }

  weight = total / 30000.0;

  cout << total << " " << weight << endl;
  return 0;
}
```

## 31.7 How to Add User-Entered Values to a One-Dimensional Array

There is nothing new here. Instead of reading a value from the keyboard and assigning that value to a variable, you can directly assign that value to a specific array element. The next C++ program prompts the user to enter the names of four people, and assigns them to the elements at index positions 0, 1, 2, and 3, of the array `names`.

### *Exercise 34.1-5 Creating Two Arrays – Separating Positive from Negative Values*

✎ *Note that the arrays* `pos` *and* `neg` *contain a total number of* `pos_index` *and* `neg_index` *elements respectively. This is why the two last loop control structures iterate until variable* `i` *reaches values* `pos_index - 1` *and* `neg_index - 1`*, respectively, and not until* `ELEMENTS - 1`*, as you may mistakenly expect.* ~~*Obviously the sum of* `pos_index + neg_index` *equals to* `ELEMENTS`.~~