Name: _____

1. (1 pt.)
   - **Read all material carefully.**
   - *If in doubt whether something is allowed, ask, don't assume.*
   - You may refer to your books, papers, and notes during this test.
   - E-books may be used *subject to the restrictions* noted in class.
   - Computers are not permitted, except when used strictly as e-books or for viewing ones own notes.
   - Network access of any kind (cell, voice, text, data, ...) is not permitted.
   - Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
   - Use class and textbook conventions for notation, algorithmic options, etc.
   - Do not attach or remove any pages.

   Write your name in the space provided above.

   Do not write on this page below this point.

WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.

Do not write on this page.
(It is for use in grading only.)

Q	Full	Score
1	1	
2	2	
3	2	
4	2	
5	2	
6	6	
7	5	
8	10	
total	30	

2. (2 pts.) Provide a single C++ statement that declares an array, named `a5`, of five unsigned integers and initializes it to contain the elements (in index order): 3, 1, 4, 1, 5.

3. (2 pts.) Provide a single C++ statement that prints, to *standard output*, the number of elements (items) in an array named `howMany`, whose elements are of type `float`.

4. (2 pts.) Provide a single C++ statement that declares an array, named `hislah`, containing four elements of type `char`, and initializes it to contain the elements (characters, in index order): `y`, `e`, and `s`.

5. (2 pts.) Provide a single C++ statement that has the same effect as the one in Question 4, and that is *as different from that one as possible*.

6. (6 pts.) Describe, as precisely as possible, the output produced by the following C++ program. (If the program will not compile, will crash, or otherwise not produce output, then explain why clearly.) *Explain the reason for your predicted output as precisely as possible; there is no credit for answers without explanations.*

```
#include <iostream>
int x[3];
int main() {
  int y[3], z[] = {1, 2, 3};
  for(int i=1; i < 3; i++) y[i] = z[i-1];
  for(int i=0; i < 3; i++) cout << 100*x[i] + 10*y[i] + z[i] << endl;
  return 0;
}
```

7. (5 pts.) Provide well-formatted C++ code that defines a function `rot3` that cyclically assigns each of its arguments to have the value of the next one. In more detail, the definition should ensure that the following program, which uses but does not define `rot3`, produces "1 4 3" as output.

```
#include <iostream>
// your rot3 definition goes here
int main() {
  int a = 3, b = 1, c = 4;
  rot3(a, b, c);
  std::cout << a << " " << b << " " << c << std::endl;
  return 0;
}
```

8. (10 pts.) Provide **well-formatted source code of a complete C++ program** that:

   1. Prints a prompt "Integers: " to *standard error* (note, not standard output).
   2. Reads five whitespace-separated integers from *standard input*. Note that the amount of separating whitespace is arbitrary and may include spaces, tabs, newlines, etc.
   3. Stores what is read in a suitably named and defined *array* variable.
   4. Writes six lines to *standard output*. The first five lines consists of the five integers read earlier, in input order and one per line. The last line consists of the sum of those integers.

   Poorly formatted, messy, or otherwise hard to read code will result in very substantial loss of points. *Explain your answer briefly, especially to qualify for partial credit.*