# AP® Computer Science A
# 2003 Scoring Commentary



These materials were produced by Educational Testing Service® (ETS®), which develops and administers the examinations of the Advanced Placement Program for the College Board. The College Board and Educational Testing Service (ETS) are dedicated to the principle of equal opportunity, and their programs, services, and employment policies are guided by that principle.

The College Board is a national nonprofit membership association whose mission is to prepare, inspire, and connect students to college and opportunity. Founded in 1900, the association is composed of more than 4,300 schools, colleges, universities, and other educational organizations. Each year, the College Board serves over three million students and their parents, 22,000 high schools, and 3,500 colleges through major programs and services in college admissions, guidance, assessment, financial aid, enrollment, and teaching and learning. Among its best-known programs are the SAT®, the PSAT/NMSQT®, and the Advanced Placement Program® (AP®). The College Board is committed to the principles of equity and excellence, and that commitment is embodied in all of its programs, services, activities, and concerns.

For further information, visit www.collegeboard.com



For the College Board's online home for AP professionals, visit AP Central at apcentral.collegeboard.com.

# AP® COMPUTER SCIENCE A
# 2003 SCORING COMMENTARY

## Question 1

Sample A(Score 9)

This solution is completely correct and was the most common solution we found. It used a `for-loop` in part (a) and an empty vector, resized by 1 each time a `College` was added in part (b).

Sample B(Score 9)

This solution is completely correct. Part (b) shows a second common algorithm for this question. It initialized the result vector to the same size as `myColleges`, then appropriately resizing the result vector at the end.

Sample C(Score 6)

This solution was actually the most commonly-found 6-point solution. The solution received full credit on part a, but contained several errors in part b. All losses were due to confusion about how to add items to an `apvector`. This solution lost ½ point for declaring an empty vector without a later resize, ½ for lack of a counter to keep track of position in the result vector, 1 for incorrectly adding to vector (and trying to add just the name, rather than the `College` object), and ½ for incorrect size of the result vector.

## Question 2

Sample A(Score 9)

In part (a) the student counted the number of satisfied conditions and returned a boolean based on that count. In part (b), an eligible employee is removed by shifting the remaining employees toward the beginning of the vector and reducing the size of the vector by 1. The loop control variable is adjusted to process the newly shifted employees. A separate loop accumulates the salaries of the remaining employees.

Sample B(Score 8)

Part (a) has a correct, though lengthy, solution. It looks for pairs of satisfied conditions. Part (b) is nearly correct. It uses an auxiliary vector to store the remaining employees. An employee that is retained is placed into the auxiliary vector and its size is increased by 1. The salary of a retiring employee is subtracted from salaryBudget. At the end, the auxiliary vector is copied back into empList. The student lost the resize point because the resulting vector contained one too many elements.

### Question 2 (cont'd.)

Sample C(Score 6)

This is a typical student solution. On part (a) the student only checks the single case when all 3 conditions are satisfied, thus losing 2 out of 3 points. On part (b), the student loses 1 point because the shift array method skips consecutive eligible employees.

### Question 3

Sample A(Score 9)

In part(a), this student checked for row and column in bounds in all directions and for a treasure in the (row, col) position sent in and returned the `boolean` value. Part(b) is an example of the nested loops solution in which a three-by-three block is examined and then 1 is subtracted since the center of the block has a treasure. Part(c) is completely correct.

Sample B(Score 9)

Part(a) demonstrates a common variation of a correct response. Part(b) is an example of the solution that correctly examines the eight positions neighboring the position (row, col). Part(c) represents the most frequently seen correct response.

Sample C(Score 7)

In part(a), this student examined the position at row, col for the presence of a treasure but failed to check row and/or col against some boundary. Both checks needed to be included for an attempt, so this student loses both half-points. In part(b), the student does not call `HasTreasure` and does not correctly re-implement `HasTreasure`, losing both `HasTreasure` half-points. Part(c) is correct. The conditional statement `theMap.HasTreasure = True` has two non-penalized usage errors: `=` instead of `==` and `True` instead of `true`.

# AP® COMPUTER SCIENCE A
# 2003 SCORING COMMENTARY

## Question 4

Sample A(Score 8)

This solution is nearly perfect. Part (c) lost two half points for missing calls to the `Update` function.

Sample B(Score 8)

Part (a) is correct. Part (b) lost one half point for initialization. To be correct the maximum should be initialized to negative one, or the test should be for greater than or equal to one to assure that some position in `nbrs` is located.

Sample C(Score 7)

Part (a) and part (b) are correct. Part (b) is an example of a canonical solution. Part (c) lost a half point (the else/logic point) for incorrect organization. Since the student did not follow the given algorithm a fish could die before being given a chance to eat. This solution also lost half points in part (c) for missing calls to update and an incorrect test for the third step since eating.