CS 1570 – Fall 2016
HW #1 Solution
Note: Other solutions are possible

Problem 1

13 pts.:

- **Steps of algorithm are clearly stated (5 pts.)**
- **Correctly gets each student somehow "associated with" his/her ID (8 pts.)**

In this solution assume:
N = # of students, studentOwnID[i] = student i's ID, studentHasID[i] = ID that student i is holding, i = 1..N

1. **currentStudent = 1**
2. **If currentStudent == N then go to 11**
3. **nextStudent = currentStudent**
4. **If studentOwnID[currentStudent] == studentHasID[nextStudent] then go to 7.**
5. **nextStudent = nextStudent + 1**
6. **Go to 4.**
7. **studentHasID[nextStudent] = studentHasID[currentStudent]**
8. **studentHasID[currentStudent] = studentOwnID[currentStudent]**
9. **currentStudent = currentStudent + 1**
10. **Go to 2.**
11. **All students leave // optional to say this ;)**

Problem 2

15 pts.:

- **Steps of algorithm are clearly stated (5 pts.)**
- **Prompts user for input (1 pt.)**
- **Correctly computes base 10 conversion (8 pts.)**
- **Outputs result (1 pt.)**

1. **Prompt user to enter a number in base 12**
2. **base12Num = user's input // a sequence of chars ('0'..'9', 'A', 'B'); 1-based indexing**
3. **n = length of base12Num**
4. **base10Num = 0 // an integer**
5. **factor = 1**
6. **If N == 0 then go to 22.**
7. **If base12Num[n] == '0' then base10Num = base10Num + (factor * 0)**
8. **If base12Num[n] == '1' then base10Num = base10Num + (factor * 1)**
9. **If base12Num[n] == '2' then base10Num = base10Num + (factor * 2)**
10. **If base12Num[n] == '3' then base10Num = base10Num + (factor * 3)**
11. **If base12Num[n] == '4' then base10Num = base10Num + (factor * 4)**
12. **If base12Num[n] == '5' then base10Num = base10Num + (factor * 5)**
13. **If base12Num[n] == '6' then base10Num = base10Num + (factor * 6)**
14. **If base12Num[n] == '7' then base10Num = base10Num + (factor * 7)**
15. **If base12Num[n] == '8' then base10Num = base10Num + (factor * 8)**

16. If base12Num[n] == '9' then base10Num = base10Num + (factor * 9)
17. If base12Num[n] == 'A' then base10Num = base10Num + (factor * 10)
18. If base12Num[n] == 'B' then base10Num = base10Num + (factor * 11)
19. n = n – 1
20. factor = factor * 12
21. Go to 6.
22. Output base10Num

Problem 3

22 pts.:

- Steps of algorithm are clearly stated (5 pts.)
- Prompts user for input of the 2 words (1 pt.)
- Handles 2 empty strings (2 pts.)
- Handles 1 empty and 1 non-empty string (1 pt.)
- Handles 2 identical strings (2 pts.)
- Handles 2 different strings of same length (2 pts.)
- Handles 2 different strings where 1st string is longer than 2nd (2 pts.)
- Handles 2 different strings where 2nd string is longer than 1st (2 pts.)
- Handles 2 different strings that differ on first letter (2 pts.)
- Handles 2 different strings that differ on something other than 1st letter (2 pts.)
- Outputs result (1 pt.)

In this solution assume:
abs(x) gives the absolute value of integer x, ascii(ch) gives the ASCII value of character ch, length(w) gives the length of a null-terminated sequence of characters w (not counting the null character)

1. Prompt user to enter first word
2. w1 = user's input for first word // a sequence of chars; 1-based indexing
3. Prompt user to enter second word
4. w2 = user's input for second word // a sequence of chars; 1-based indexing
5. n1 = length of w1
6. n2 = length of w2
7. If n1 > 0 and n2 > 0 then go to 11.
8. If n1 == 0 and n2 > 0 then lexDiff = ascii(w2[1])
9. If n1 > 0 and n2 == 0 then lexDiff = ascii(w1[1])
10. Go to 19
11. lexDiff = 0
12. i = 1
13. If i <= n1 and i <= n2 then lexDiff = abs(ascii(w1[i]) – ascii(w2[i]))
14. If i <= n1 and i > n2 then lexDiff = ascii(w1[i])
15. If i > n1 and i <= n2 then lexDiff = ascii(w2[i])
16. If lexDiff > 0 then go to 19.
17. i = i + 1
18. Go to 13.
19. Output lexDiff