

Homework 5

PROBLEM SOLVING IN C
(CSCE 105, FALL 2005)

URL: <http://www.cse.unl.edu/~cstrobe/csce105f05/>

Due on 2nd December, 2005

14th November, 2005

Name :
Course No : **CSCE105**

Instructions:

All of the questions 1...8 are programming projects. For each problem, turn in your source code using webhandin, naming your files <user-id_#.c>, where user-id is your CSE login name, and '#' is the problem number (between 1...8). **If you use a compiler other than CSE, make sure that your code works on CSE!**

1. (10 points) Write a program that will compute and display both the range of values and the standard deviation for a collection of N numbers. To compute the standard deviation, accumulate the sum of the squares of the data values (`sum_squares`) in the main loop. After loop exit, use the formula

$$\text{standard deviation} = \sqrt{\frac{\text{sum_squares}}{N} - \text{average}^2}.$$

Get the value of N before scanning (using `scanf`) each value in the collection of N numbers.

2. (10 points) The rate of decay of a radioactive isotope is given in terms of its half-life H , the time lapse required for the isotope to decay to one-half of its original mass. The isotope strontium-90 (^{90}Sr) has a half-life of 28 years. Compute and print in table form the amount of this isotope that remains after each year for n years, given the initial presence of an amount in grams. The values of n and *amount* should be provided interactively. The amount of ^{90}Sr remaining can be computed by using the following formula:

$$r = \text{amount} \times C^{(y/H)}$$

where *amount* is the initial amount in grams, C is expressed as $e^{-0.693}$ ($e = 2.71828$), y is the number of years elapsed, and H is the half-life of the isotope in years.

3. (10 points) Since communication channels are often noisy, numerous ways have been devised to ensure reliable data transmission. One successful method uses a checksum. A checksum for a message can be computed by summing the integer codes of the characters in the message and finding the remainder of this sum divided by 64. The integer code for a space character is added to this result to obtain the checksum. Since this value is within the range of the displayable characters, it is displayed as a character as well. Write a program that accepts single-line messages ending with a period and displays the checksum character for each message. Your program should continue displaying checksums until the user enters a line with only a period.

Hint: Since each line ends with the character '.', you can assume that the '.' is a sentinel character.

4. (10 points) Let `arr` be an array of 20 integers. Write a program that first fills the array with up to 20 input values and then finds and displays both the *subscript* of the largest item in `arr` and the value of the largest item.
5. (10 points) Write an interactive program that plays a game of hangman. Store the word to be guessed in successive elements of an array of individual characters called `word`. The player must guess the letters belonging to `word`. The program should terminate when either all letters have been guessed correctly (the player wins) or a specified number of incorrect guesses have been made (the computer wins).

Hint: use another array, `guessed`, to keep track of the solution so far. Initialize all elements of `guessed` to the '' symbol. Each time a letter in `word` is guessed, replace the corresponding '*' in `guessed` with that letter.*

6. (10 points) Write a program that takes nouns and forms their plurals on the basis of these rules:
- a. If noun ends in “y,” remove the “y” and add “ies.”
 - b. If noun ends in “s,” “ch,” or “sh,” add “es.”
 - c. In all other cases, just add “s.”

Print each noun and its plural. Try the following data:

chair dairy boss circus fly dog church clue dish

7. (20 points) Write a program that processes a sequence of lines, displaying a count of the total number of words in those lines as well as counts of the number of words with one letter, two letters, and so on.
8. (20 points) Write a program that takes data a line at a time and reverses the words of the line. For example,

Input: birds and bees

Reversed: bees and birds

The data should have one blank between each pair of words.

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	20	
8	20	
Total:	100	