

CSCE 121 Assignment #3

due September 22 to eCampus

Objectives: This assignment gives you some experience with writing C++ programs that use fundamental operations: loops, comparisons, functions, and vectors. You will also have an opportunity to write code that catches errors and throws exceptions.

All work should be pre-graded in the labs and turned in to eCampus by the deadline. For programs, you need to turn in only the source code (not object or executable files). Your code will be tested using the g++ compiler; you can develop your program in Visual Studio or Xcode, but please make sure your code also compiles and runs on a Linux machine.

1. (5 points) Create a text file, called README, in which you provide:

- Your first name, last name, UIN, section number, user name, e-mail address
- State the Aggie Honor statement:

I certify that I have listed all the sources that I used to develop the solutions/code to the submitted work.

On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.

Your Name

Date

- List any resources used, outside the textbook and discussions with the Instructor, TA, or Peer Teachers, such as webpages titles and URLs and other sources. Describe in details the type of help you have obtained from each resources.
 - List any known problems with the assignments you are turning in. For example, if you know your code does not run correctly, state that. There should be a short explanation.
 - Submit an electronic version of the README file along with C++ source files for the problems listed below and a hard copy to your TA.
2. (Lab problem 6 points) Modify the C++ program to the problem 5 from the assignment 2 by introducing the function:

```
void print_pyramid(int n)
```

where `n` is the number of rows. Call this function from `main()` to perform testing for 5 different values of `n`. How can you report an error when the parameter `n` is not in the range between 1 and 15.
 3. (Lab problem 24 points) The minimum function.
 - (a) Write a function that takes two integers and returns the value of the smaller one. In the `main()` function provide 5 test cases to verify its correctness.
 - (b) Write the function that takes two characters and return the smaller one in the lexicographical order. Write the `main()` function that tests that functions for 5 different pairs of character type variables.
 - (c) Write a generic function that takes two numeric objects and returns the value of the smaller one according to the less-than operation. Test the function in `main()` using the integer and character types.
 4. (40 points) Write a C++ function for computing the solution to a quadratic equation $ax^2 + bx + c = 0$. Use the quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. If $b^2 - 4ac < 0$, report that there are no real roots (there are two complex roots). Consider the special case when $a == 0$ when the quadratic equations becomes a linear equation.

- (a) Write a function for computing the roots using the following function header:

```
int compute_roots(double a, double b, double c,
                 double& x1, double& x2)
```

The arguments a, b, c are the coefficients of the quadratic equation, and $x1$ and $x2$ are possible computed roots of the equation (depending on the return value). The return value of this function could be one of the numbers:

0 == no roots

1 == one root

2 == two real roots

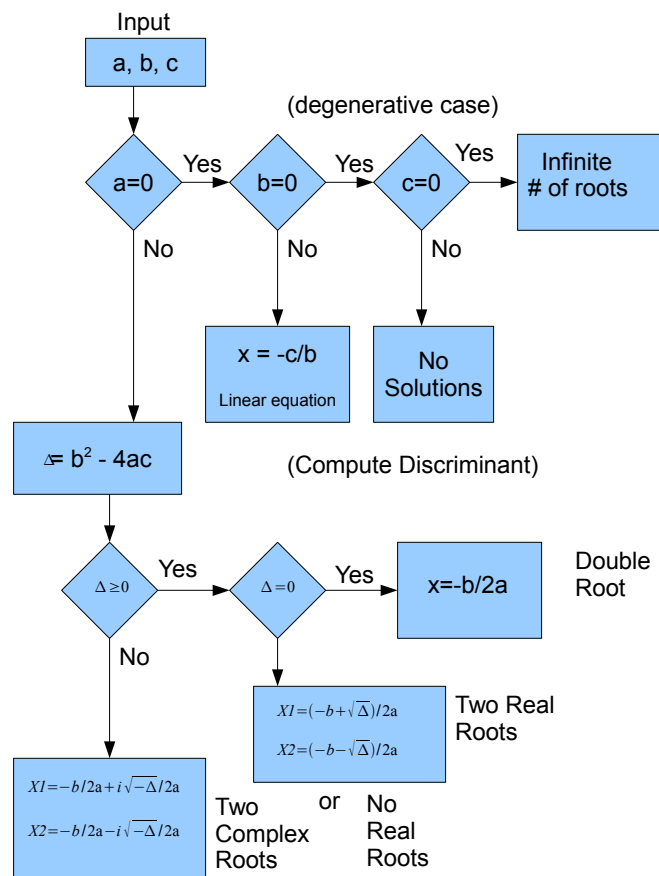
3 == infinite number of roots

4 == no real roots

You should save the return value in a variable, say, `nroots` and use it later in a switch statement `switch(nroots)`. Do not include any input or output statement inside `compute_roots()`. All input/output statements should be in `main()` only.

- (b) Test your function in `main()` for five different quadratic equations. Use the included flow chart when writing the function.

Flowchart for $ax^2 + bx + c = 0$



- (c) Write in the README file about what you have learned by writing this assignment and what results of your testing are. Write also about your (possible) errors or mistakes.

5. (55 points) Implement in C++ a game called “*Your Lucky Number*”. The lucky number is calculated based on an individual’s birth month, day, and year. The algorithm assigns any person a number between 1 and 9. Many people believe that a group of people with the same lucky number share common features. The *lucky number* is calculated as the sum of digits of the birth month, day, and year numbers. If this sum is not a single digit then you need to keep adding up its digits. Use the C++ functions when you implement the game. Test your program for corrections.

To start the game you should enter your full birthdate in the format: month day year, e.g. 9 21 1985. The month is represented by its corresponding integer: January is 1, February is 2, ..., and December is 12. Be sure that your program will verify the input format and print an error message when the format is incorrect (use exceptions), and will ask to re-enter the birthdate again.

The algorithm for calculating the lucky number

- (a) Enter the month as an integer from 1 to 12.
- (b) Reduce the month, day and year numbers down to a one-digit number by adding all digits in month, day and year numbers, see the example below.
- (c) Map the number to the category listed below and display the message on the screen with at least four features for a person in each category. I provided only one feature (as an adjective) for each category and you should complete the list.
- (d) Test your program using at least 10 your friends’ or relatives’ birthdays.
- (e) Provide statistics in the README file about the results of your testing by giving the percentage of people that confirm that they belong to the category calculated by your program. Write in the README file about what you have learned by writing this assignment. Write also about your (possible) problems, errors or mistakes.

Categories:

1. The Leader: original thinker (add 3 more features)
2. The Mediator: diplomatic (add 3 more features)
3. The Communicator: expressive (add 3 more features)
4. The Teacher: trustworthy (add 3 more features)
5. The Freedom Seeker: adventurous (add 3 more features)
6. The Nurturer: family-oriented (add 3 more features)
7. The Seeker: analytic (add 3 more features)
8. The Power House: authoritative (add 3 more features)
9. The Humanitarian: charitable (add 3 more features)

Example

The lucky number for the date “September 21 2012” is 8, see the calculations below.

- The month digit is 9 because September is 9th month of the year.
- The day number 21 reduces to 3 as a single digit because $2 + 1 = 3$
- The year number reduces to 5 as a single digit because $2 + 0 + 1 + 2 = 5$
- Add up the above three numbers: $9 + 3 + 5 = 17$, and continue adding digits to reduce it to a single digit: $1 + 7 = 8$
- The lucky number is 8 and it falls into the category Power House.