TEAM INFORMATION FOR ALL-REGION COMPETITION DAY

Email Submission Setup

Submission Email:

- Once the team completes the competition packet, it is their responsibility to email that packet, addressed to **BOTH** CSforAR@ade.arkansas.gov and CSforAR@gmail.com, by 1:00 p.m. of the competition day.
- The official time of receipt will be recorded as the earliest time that the packet is received at either email address.
- The team will attach all files (the multiple-choice answers text file and all challenge files) for submission to be judged to these emails.
- The email should contain the school name with team number identifier (if applicable) and team member abbreviations followed by "sub" in the subject line and once again in the body of the email. Each word or initial should be separated by an underscore.
- A scoring penalty of 5%, of the maximum allowable score, will be deducted from the final team score for each minute past 1:00 p.m. that the packet is received. Any packets received after 1:20 p.m. of the competition day will automatically be disqualified.

Example Email Submission Subject Line Process:

- School name: Duck High School 1
- Team Member Names: Donald Mouse, Willy Fudd, Optimus Flintstone
- Email Submission Subject Line: **Duck_High_School_1_DM_WF_OF_sub**

Multiple-Choice Questions

File and Name: Answers to all multiple-choice questions must be submitted in one plain text file titled MQA.txt.

Documentation: MQA.txt should include the team's school name and all team member names at the top of the file.

Answer Format: Each answer should be on a new line preceded by the number and a period then the answer in upper-case format.

Example:

```
1. A
2. B
3. C
…
```

Scoring: Teams should select the best single answer to the following questions from the choices provided; there is no additional penalty for wrong answers. The lowest score a team may receive for any question is zero. Questions with more than one answer or no answer will be marked as incorrect. The multiple-choice section will be calculated as 20% of the team's overall score.

Challenges

Files: Each separate challenge should be attached to the team's submission email, as **a separate single file** as indicated in the "Solution Language" section of the Competition Rules. Only one file per challenge will be accepted.

Names: Each challenge should be titled with the team's initials, underscore, the word "challenge", the challenge number, and the appropriate extension for your solution language. **See the example below.**

Example:

- Challenge: 2
- Team Member Names: Donald Mouse, Willy Fudd, Optimus Flintstone
- Solution Language: Java
- File name would be: dmwfof_challenge2.java

Documentation: Each challenge file should include the team's school name and all team member names in the documentation at the top of the source code.

Scoring of Coding Challenges

Program functionality is most important. Your team may get **partial points** for being able to complete some of the listed tasks for each challenge. Points may also be awarded based on teams following good programming practices, including but not limited to, appropriate commenting/documentation, consistent indentation, appropriate code grouping, consistent/appropriate naming, error checking, and efficient code.

Teams may use any development environment or editor to test their solutions. However, keep in mind that the judges may not use the same development environment or editor as the team.

The challenges section will be calculated as 80% of the team's overall score.

Final Scores

Teams should attempt to answer as many questions and challenges as possible. There is no penalty for guessing. The lowest score a team may receive for any question or challenge is zero. The questions and challenges are not pass or fail. **The top teams are determined by having the highest scores in relation to all teams**.

A scoring penalty of 5%, of the maximum allowable score, will be deducted from the final team score for each minute past 1:00 p.m. that the packet is received. Any packets received after 1:20 p.m. of the competition day will automatically be disqualified.

Errors Within This Document

If a team locates what is perceived to be an error in the document, the team may email CSforAR@ade.arkansas.gov and CSforAR@gmail.com. However, no questions will be altered or given further explanation during the competition time.

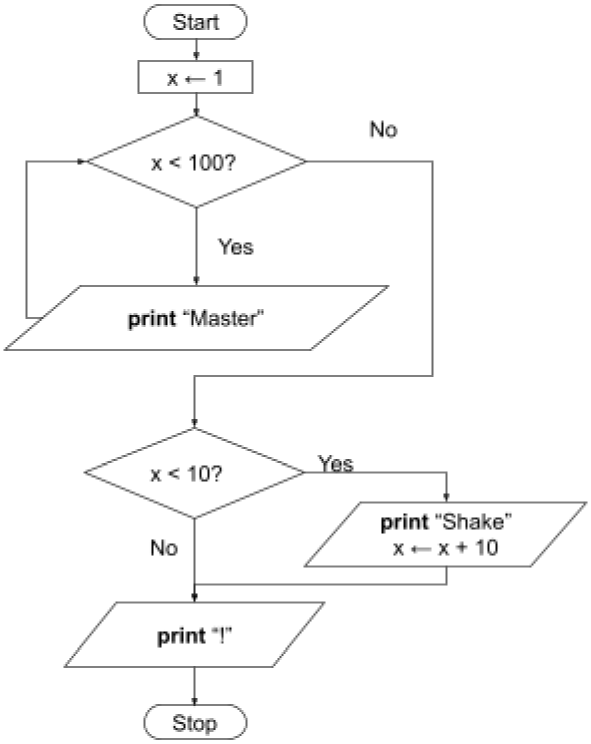
Rules

Teams should have access to a copy of and are responsible for following the full set of rules.
http://bit.ly/ARCSHistCompRules

Multiple Choice Section (30 Questions)

Instructions: *Each answer should be on a new line preceded by the number and a period; then the answer in upper-case format. See the above instructions for more information. Example:*
1. A
2. B
3. C
…

1. Consider the following flowchart.



How many times will "Master" and "Shake" print?

a. Both "Master" and "Shake" will print 10 times.
b. "Master" will print 1 time and "Shake" will print 10 times.
c. "Master" will print 100 times and "Shake" will print 10 times.
d. "Master" will print an infinite amount of times and "Shake" printing 0 times.

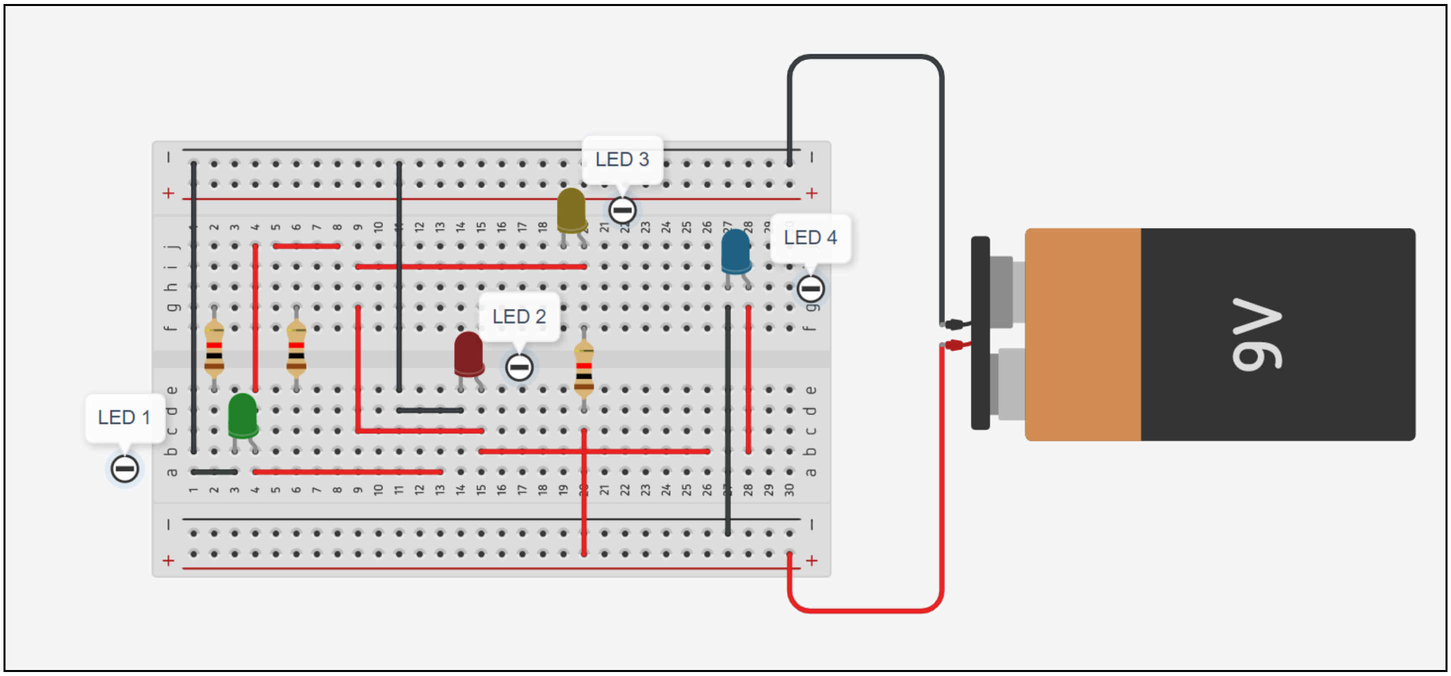
2. What is the average-case runtime of the following **cool_func** procedure?

```
// precondition 1: A is an array of integers.
// precondition 2: The length of array A is n.
// precondition 3: The index of array A starts at 0.

int [ ] cool_func ( int [ ] A, int n )
     for ( int j = 1; j ≤ n - 1; j = j + 1 )
          int temp = A [ j ]
          int k = j - 1
          while  ( ( k ≥ 0 ) and ( A [ k ] > temp ) )
               A [ k + 1 ] = A [ k ]
               k = k - 1
          A [ k + 1 ] = temp
     return A
```

   a. linear
   b. logarithmic
   c. quadratic
   d. exponential

3. Consider the following diagram of a circuit.



Which of the light-emitting diodes (LEDs) will light up?

   a. LED 1
   b. LED 2
   c. LED 3
   d. LED 4

4. A team has been tasked with finding patterns in 20 years worth of weather data for forecasting future weather events. Which machine learning technique would be most appropriate?

    a. supervised classification
    b. unsupervised classification
    c. semi-supervised classification
    d. regression analysis

5. How many IP addresses can be assigned to hosts in a Class B or CIDR /16 range?

    a. 254
    b. 16,382
    c. 65,534
    d. 4,294,967,294

6. Communication over a network generally takes place via sockets. There are two ends for every socket; the source and the destination. What role are you most likely fulfilling when you post a photo to social media?

    a. client
    b. server
    c. proxy
    d. peer

7. You are on a website with an image carousel that is constantly changing pictures. Which of the following technologies is most likely responsible for the dynamic content on the website?

    a. JavaScript
    b. HTML
    c. Cascading Style Sheet
    d. Extensible Markup Language

8. You are at your bank's login page. Below is a snippet of the PHP/SQL code that checks the validity of the user-provided credentials from the login page.

```
$sql = “select * from users where username = ‘$form_username’ and password = ‘$form_password’”;
```

Which of the following inputs will bypass the need for a password?

a. Submitting `‘ or 1=1 --` in the form_username field

b. Submitting `or 1 != 1 --` in the form_password field

c. Submitting `admin ’` in the form_username field

d. Submitting `‘–` in the form_username field

9. Consider the following pseudocode procedure.

```
int [ ] cool_func ( int m, int n )
     m = m + ( m + n ) % 2
     if ( m > n )
          return 10
     else
          return  m + cool_func( m + n % 5, n - 1 )
```

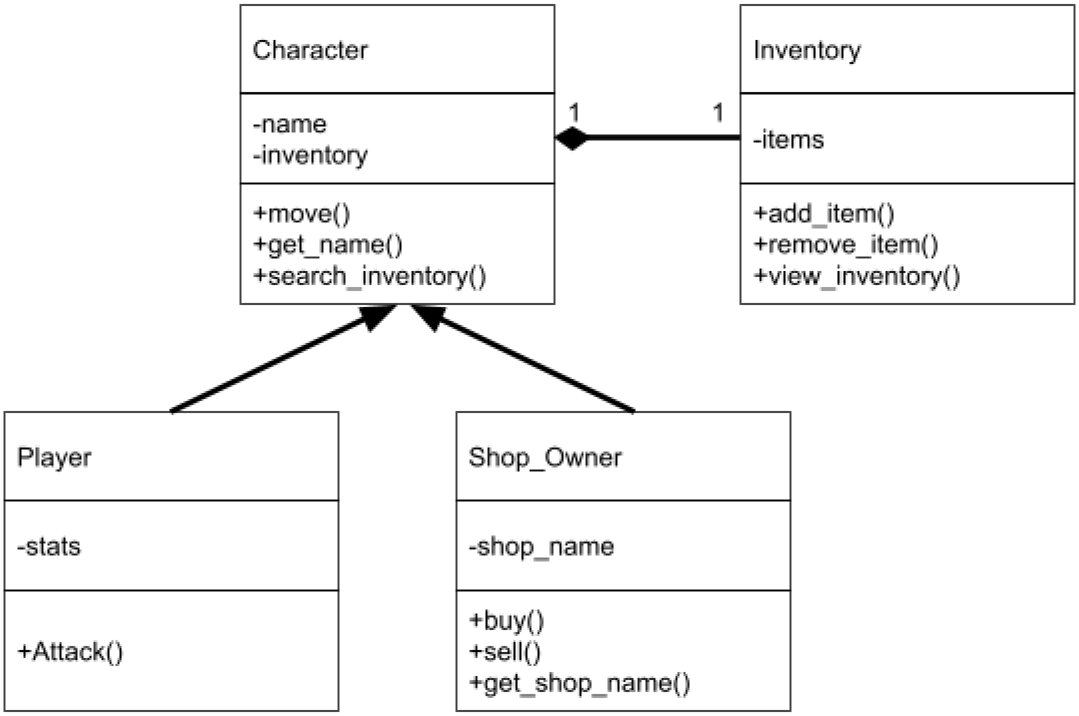
What would be returned when **`cool_func( 1, 10 )`** is executed?

   a. 2
   b. 10
   c. 23
   d. An error would occur.

10. A team was wanting to create a game manager for their next AAA game using object-oriented programming. There should only be one game manager at any given time in the game. The game manager will prevent additional game manager objects from being initialized. Which design pattern most accurately describes this requirement?

   a. Factory
   b. Singleton
   c. Observer
   d. One-off

11. Consider the following unified modeling language (UML) diagram.



The Character class containing an instance of an inventory object is an example of which of the following concepts?

a. Overriding
b. Composition
c. Inheritance
d. Overloading

12. Which of the following is the highest level of abstraction regarding a video being streamed to a screen?

a. Each individual frame (image) is displayed on your screen for playing the video.
b. Each individual pixel on your screen when the video is playing.
c. The byte sequence in a file on your computer for playing a video.
d. The video that is streaming on your screen.

13. Consider the following documentation for the implementation of a String class.

Procedure	Documentation
count()	Returns the length of a string.
upper()	Returns a string with all uppercase characters from a provided string.
lower()	Returns a string with all lowercase characters from a provided string.
substring(begin, end)	Returns a substring of characters from position first to position last in a provided string. Both positions are inclusive.

Consider the following pseudocode.

```
String cheese = “mozzarella”
String bean = cheese.substring(0, cheese.count() - 5).upper()
String theory = cheese.substring(6, cheese.count() - 1)
String y = bean + theory.substring(0, 0) + theory.substring(1, 2).upper()
String y = y.substring(0, 1).lower() + y.substring(2, y.count() - 1)
print y
```

What is the output of the pseudocode displayed above?

a. moZZAReLLA
b. mZZAe
c. moZZAReLL
d. mZZARella

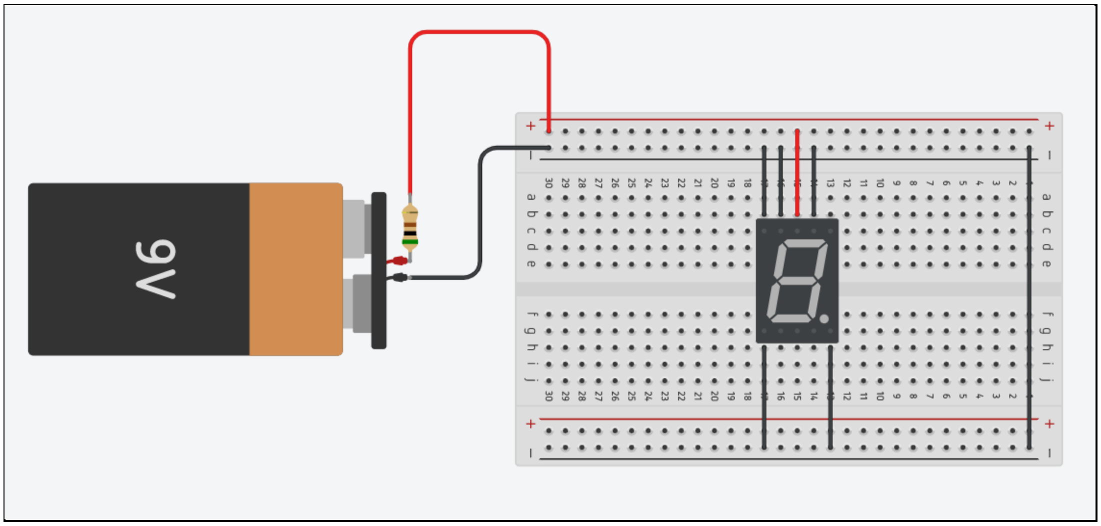
14. Consider the following pseudocode of a procedure. The variable **nums** refers to a linear data structure of integers. The variables **m** and **n** refer to indices of two values within the linear data structure. Indexing into the data structure is not influenced by the number of values stored in it.

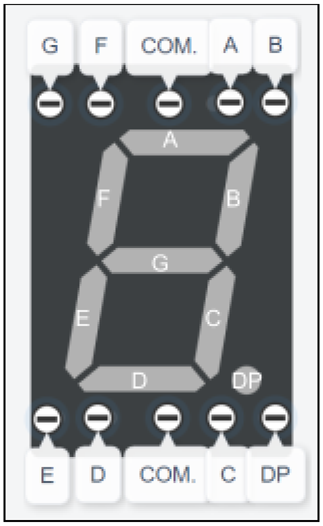
```
void cool_func ( int[] nums, int m, int n)
     int t = nums[ m ]
     nums[ m ] = nums[ n ]
     nums[ n ] = t
```

What is the runtime of the above procedure?

a. constant
b. linear
c. quadratic
d. logarithmic

15. Consider the following circuit diagram and the accompanying pinout diagram of a seven-segment display.

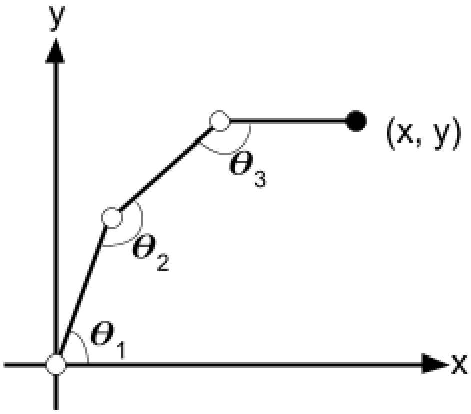




Which of the following outputs is displayed on the seven-segment display?

a. 5
b. b.
c. 6
d. F.

16. Consider the following diagram that represents a robotic arm where the black dot (x, y) is the end effector and the base of the arm is at the origin (0, 0). The angles ($\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_3$) reflect the rotation of the joints in the arm.



To determine the final joint angles using inverse kinematics, which of the following will you need to know first?

a. (x, y)
b. $\boldsymbol{\theta}_1$
c. $\boldsymbol{\theta}_2$
d. $\boldsymbol{\theta}_3$

17. A development team has created an application that allows users to connect using a URL in a popular web browser (e.g. Safari). This application would best be classified as which of the following?

    a. Native application for iOS platform
    b. Native application for Android platform
    c. Web application
    d. Hybrid application for the iOS platform

18. Morse code originally was a way to transmit information over a line using a series of dots and dashes which correspond to short and long beeps. The information was not intended to be hidden from any parties.

    ```
    hello == .... . .-.. .-.. ---
    ```

    Sending a message through morse code implements which of the following processes?

    a. Encoding
    b. Encryption
    c. Both
    d. Neither

19. A single user has intentionally crashed a program that was running on a network for all users to access. Which of the following most closely resembles the scenario described?

    a. Denial of Service attack
    b. Buffer Overflow exploit
    c. Distributed Denial of Service attack
    d. Brute Force attack

20. Your company was hired to conduct a penetration test for a local accounting firm and provide a report of your results so they can establish mitigations. Which is most likely to be outside of the scope of the penetration test?

    a. Checking for unlocked doors to gain access to the facility after hours.
    b. Sending phishing emails and vishing calls to elicit information about the network and personnel.
    c. Attempting a SQL injection to access customer data within their database.
    d. Filing a false police report on behalf of the organization to create a distraction to gain access to the facility.

21. Perform the following bitwise operation and select the correct answer.

    ```
    01100001 XOR 101011
    ```

    a. `00100001`
    b. `01101011`
    c. `10010100`
    d. `01001010`

22. What is the base$_8$ value 534 in base$_{16}$?

    a. F16
    b. EA7
    c. 15G
    d. 15C

23. A file contains 2.6 gigabytes of information. Approximately, how long will it take to transfer the file to a hard disk via USB at 480 megabits per second?

    a. 44 seconds
    b. 5.4 seconds
    c. 0.005 seconds
    d. 1,248 seconds

24. After training a model for 100 epochs and validating it with a large set of sample data, you now have a model with an accuracy score of 96% and a loss score of 0.4. When you begin feeding the model new data, your accuracy drops close to 50% and your loss is very high. What is most likely wrong with the model?

    a. The model is overfitted.
    b. The model is underfitted.
    c. The model has high variance.
    d. The model has a high bias.

25. If `p == True` and `q == False`, which of the following will evaluate the same as the following expression?

    ```
    not p or q and p and p or not p
    ```

    a. `not q or q and q and p or not p`
    b. `q`
    c. `not q`
    d. `not q or q and q and q or q`

26. Consider the following pseudocode of a procedure, where **houses** is a linear data structure of **House** objects. The method **`length()`** returns the number of elements inside of a linear data structure.

```
void magic_painter ( House[] houses )
     Color[] colors ← [Color.blue, Color.yellow, Color.red]
     int i = 0
     while i < houses.length()
          *** MISSING CODE ***
          i = i + 1
```

What would the *** MISSING CODE *** most likely have to be for all **House** objects in the **houses** data structure to be painted a color?

a. `House.paint_houses( colors[ i % colors.length() ] )`
b. `House.paint_houses( colors[ i ] )`
c. `houses[ i ].paint_house( colors[ i ] )`
d. `houses[ i ].paint_house( colors[ i % colors.length() ] )`

27. A game design studio is trying to determine which kind of data their game will be leveraging. They will be storing information for user accounts for an MMORPG (e.g. active, inactive, player statistics). User accounts can be added, removed, and changed from active to inactive and vice versa. Which of the following best describes the mode of data that they are accessing?

a. Static
b. Dynamic
c. Neither
d. Both

28. Consider the following pseudocode of the **pattern** procedure.

```
void pattern ( int idx )
     if ( idx <= 1 )
           print "." // no newline
     else
          pattern( idx - 1 )

          for ( int i = 0; i < idx; i = i + 1 )
               if ( idx % 2 == 0 )
                    String s = "."
               else
                    String s = "-"

               print s // no newline
```

Which of the following output was most likely generated from **`pattern( 6 )`**?

a. `------......------......------......------......`
b. `....---....-----......`
c. `......`
d. `.-.-.-.-`

29. Which of the following command(s) displays your current network configuration?

a. `ipconfig /all` or `ifconfig && ip route`
b. `ping 127.0.0.1`
c. `tracert 127.0.0.1` or `tracepath 127.0.0.1`
d. `nslookup -d test.com 127.0.0.1`

30. A software development team has hired an intern for a summer coding project. Unfortunately, the intern accidentally deleted all of the critical source code for the latest feature they are planning to introduce. Which kind of software is used to help recover from these types of events?

a. version control system
b. source code editor
c. debugger
d. compiler

Coding Challenge # 1 - Korben Dallas Multipass

Points = 4

Instructions

Write a program that will:

- Output your team's information to the console using the following format:

```
School: <School Name>
Team Member 1: <Team Member 1 First Name> <Team Member 1 Last Name>
Team Member 2: <Team Member 2 First Name> <Team Member 2 Last Name>
Team Member 3: <Team Member 3 First Name> <Team Member 3 Last Name>
Sponsor: <Sponsor First Name> <Sponsor Last Name>
```

- See the example output below for a complete example.
- The program should not repeat and it should execute without error.

Example Team Information (your team should use your information):

School: Academy High School
Team Member 1: Hinata Hopper
Team Member 2: Sakura Lovelace
Team Member 3: Itachi Turing
Sponsor: Kakashi Hatake

Example Output for the provided Example Team Information:

```
Line 1: School: Academy High School
Line 2: Team Member 1: Hinata Hopper
Line 3: Team Member 2: Sakura Lovelace
Line 4: Team Member 3: Itachi Turing
Line 5: Sponsor: Kakashi Hatake
```

Submissions must be in the following form:

- **xxxxxx** serves as a placeholder for your initials. (See Challenge Rules)
- C++: A single C++ source file named **xxxxxx_challenge1.cpp**, containing a main function that returns the integer zero on successful exit and non-zero in case of error.
- Java: A single Java source file named **xxxxxx_challenge1.java**, containing an appropriately named public class and public main method.
- Python3: a single python3 file named **xxxxxx_challenge1.py**
- C++, Java, and Python programs must compile/execute with no additional third-party libraries linked on build time.

Coding Challenge # 1 Rubric

- 4 points - Output the team's information in the correct format.

Coding Challenge # 2 - Dating App…Well…Sorta…

Points = 10

Narrative

Your team has been contracted to build the next-generation dating app. You will set yourselves apart by ensuring that everyone gets a date. The biggest problem on the other platforms is that people keep feeling like they're getting stood up. There are tons of support messages asking whether their date was on February 3rd or March 2nd. Your program will help clear things up by enforcing a consistent format for the dates.

Instructions

Write a program that will accept dates and output a formatted date following the below guidelines:

- Prompt the user to enter a date.
  - See example inputs in this challenge to determine the format of prompts and user input.
- Accept a date as user input on the next line using the console.
  - See example inputs in this challenge to determine the format of prompts and user input.
- The date should be validated using the following guidelines:
  - The general format of the date should be **`<DDMMMYY>.`**
  - **`DD`** is the two-digit day in the range of **`00`** to **`31`**.
  - **`MMM`** is the all-caps abbreviation for the month (**`JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC`**`).`
  - **`YY`** is the two-digit year in the range of **`00`** to **`99`** starting with the year **`2000`**.
- The program should handle incorrect input by displaying `Wrong Date Format` to the console and closing the program without an error.
- Parse the date and output the information in the following format to the console.
  - **`Day: <Day>`**
  - **`Month: <Correctly Spelled Full Month Name>`**
  - **`Year: <Four Digit Year>`**
- See the example input and output below for how the prompt, user input, and output should be formatted.
- The program should not repeat and it should execute without error.

Example Input 1

```
    Prompt: Enter Date:
User Input: 02FEB22
```

Example Output 1

```
Line 1: Day: 02
Line 2: Month: February
Line 3: Year: 2022
```

Example Input 2

```
      Prompt: Enter Date:
  User Input: February 2, 2022
```

Example Output 2

```
      Line 1: Wrong Date Format
```

Submissions must be in the following form:

- **xxxxxx** serves as a placeholder for your initials. (See Challenge Rules)
- C++: A single C++ source file named **xxxxxx_challenge2.cpp**, containing a main function that returns the integer zero on successful exit and non-zero in case of error.
- Java: A single Java source file named **xxxxxx_challenge2.java**, containing an appropriately named public class and public main method.
- Python3: a single python3 file named **xxxxxx_challenge2.py**
- C++, Java, and Python programs must compile/execute with no additional third-party libraries linked on build time.

Coding Challenge # 2 Rubric

- 2 points - The program correctly prompts and allows the user to enter a date into the program via the console.
- 2 points - The program correctly handles input that does not match the outlined format.
- 6 points - Output the correct information to the console in the correct format. See the **Example Output** for the correct format.

Coding Challenge # 3 - vIP List

Points = 16

Narrative

Your team was selected to implement the new firewall rule processing engine for a new piece of network security software. The software's job is to determine which IP addresses are allowed to communicate on the network. Before it can do that, it needs to know whether or not this is even a correct IPv6 address. No sense in blocking invalid IPs! Your team will have to figure out how to make this firewall rule processing engine come to life so the company can ship this new product.

Instructions

Write a program that will:

- Prompt the user to enter a potential IPv6 address.
  - See example inputs in this challenge to determine the format of prompts and user input.
- Accept a potential IPv6 address as user input on the next line in the console.
  - See example inputs in this challenge to determine the format of prompts and user input.
- Use the below guidelines to determine a correct IPv6 address.

IPv6 Guideline	**Example**
Eight sets of hexadecimal numbers	`FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF`
Each set has four hexadecimal digits (`0000 - FFFF`)	`A01F:0011:5CD3:E321:CCCC:FFFF:0000:0122`
Leading 0's can be dropped in each set. If there are 4 zeros in a set, one will remain after this process.	`A01F:11:5CD3:E321:CCCC:FFFF:0:122`
Sets of 0's can be omitted entirely leaving the left and right colon (::). No zeros will remain between the colons after this process.	`A01F:11:5CD3:E321:CCCC:FFFF::122`

Contiguous sets of 0's can be collapsed. When collapsing, no zeros will remain between the colons.	Before collapse:
	`FFFF:FFFF:0000:0000:0000:FFFF:FFFF:FFFF`
	After collapse:
	`FFFF:FFFF::FFFF:FFFF:FFFF`

- Correctly output `CORRECT` if the IPv6 address follows the above guidelines or `WRONG` if it does not follow the above guidelines.
  - `CORRECT` or `WRONG` should be output to the console.
- See the example input and output below for how the prompt, user input, and output should be formatted.
- The program should not repeat and it should execute without error.

Example Input 1

```
     Prompt: Enter An IPv6 Address:
 User Input: A01F:0011:5CD3:E321:CCCC:FFFF:0000:0122
```

Example Output 1

```
     Line 1: CORRECT
```

Example Input 2

```
     Prompt: Enter An IPv6 Address:
 User Input: 127.0.0.1
```

Example Output 2

```
     Line 1: WRONG
```

Example Input 3

```
     Prompt: Enter An IPv6 Address:
 User Input: A01F:11:5CD3:E321:CCCC:FFFF:0:122
```

Example Output 3

```
     Line 1: CORRECT
```

Example Input 4

```
     Prompt: Enter An IPv6 Address:
 User Input: A01F:11:5CD3:E321:CCCC:FFFF::122
```

Example Output 4

```
     Line 1: CORRECT
```

Example Input 5

```
     Prompt: Enter An IPv6 Address:
 User Input: C23:53:2::FF:3456:3D:483
```

Example Output 5

```
     Line 1: CORRECT
```

Example Input 6

```
     Prompt: Enter An IPv6 Address:
 User Input: 34.23.13456.3245H.403.7457.2.3DEFF
```

Example Output 6

```
     Line 1: WRONG
```

Additional Information

Submissions must be in the following form:

- **xxxxxx** serves as a placeholder for your initials. (See Challenge Rules)
- C++: A single C++ source file named **xxxxxx_challenge3.cpp**, containing a main function that returns the integer zero on successful exit and non-zero in case of error.
- Java: A single Java source file named **xxxxxx_challenge3.java**, containing an appropriately named public class and public main method.
- Python3: a single python3 file named **xxxxxx_challenge3.py**
- C++, Java, and Python programs must compile/execute with no additional third-party libraries linked on build time.

Coding Challenge # 3 Rubric

- 2 points - The program correctly prompts and allows the user to enter a potential IPv6 address to validate via the console.
- 14 points - The program will correctly output if the provided IPv6 address is `CORRECT` or `WRONG`. Partial bullet points may be awarded.

Coding Challenge # 4 - Base-Backwards, LLC.
Points = 14

Narrative

A base system allows for a way to represent numerical quantities. Encoding is the process of translating information so that other systems can understand it. American Standard Code for Information Interchange (ASCII) is the mapping of numeric values to a character that can be displayed on the screen. The character “A” can be represented by the decimal number 65 in and the character “Z” can be represented by the decimal number 90. Also, the character “a” can be represented by the decimal number 97 and the character “z” can be represented by the decimal number 122.

The scheme you will be implementing is called Base-16P. It’s a new proprietary scheme to encode information using existing CPU architectures for maximum lifecycle synergy optimization. This will allow for ideal transliteration between disparate anonymized systems using zero-trust architecture to facilitate erroneous behavior. Your team will be responsible for successfully decoding these Base-16P messages.

Instructions

Write a program that will decode Base-16P messages into ASCII characters following the guidelines below:

- Prompt the user to enter a number encoded using the proprietary Base-16P scheme.
  - See example inputs in this challenge to determine the format of prompts and user input.
- Accept a number encoded using the proprietary Base-16P scheme as user input on the next line in the console.
  - See example inputs in this challenge to determine the format of prompts and user input.
  - The proprietary scheme follows the guidelines below.

Guideline	Documentation
`base-16`	The encoded value will be a base-16 value. Each digit will have 16 options of characters to represent it.

`digits`	The set of digits are: F, E, D, C, B, A, 9, 8, 7, 6, 5, 4, 3, 2, 1, and 0
	A mapping each digit to base-10:
	`'F': 0    'B': 4    '7': 8     '3': 12`
	`'E': 1    'A': 5    '6': 9     '2': 13`
	`'D': 2    '9': 6    '5': 10    '1': 14`
	`'C': 3    '8': 7    '4': 11    '0': 15`

`digit pairs`	Each pair of Base-16P digits in the provided input represents a decimal value from 0-255.

- The program should handle incorrect input by displaying `Invalid Input` to the console and closing the program without an error.
- Decode the Base-16P input to ASCII characters.

    - Assume that all messages will contain only alphabetic [a-zA-Z] ASCII characters.
- Output the ASCII characters to the console.
- See the example input and output below for how the prompt, user input, and output should be formatted.
- The program should not repeat and it should execute without error.

Example Input 1

```
     Prompt: Proprietary Scheme:
 User Input: BCAC99908DBEAD
```

Example Output 1

```
     Line 1: CSforAR
```

Example Input 2

```
     Prompt: Proprietary Scheme:
 User Input: B6ABA8B0ADB4ACB0B1ABB7B6ACB2BEBCB7B6B1BA
```

Example Output 2

```
     Line 1: ITWORKSONTHISMACHINE
```

Example Input 3

```
     Prompt: Proprietary Scheme:
 User Input: 979E9C948B979A8F939E919A8B
```

Example Output 3

```
     Line 1: hacktheplanet
```

Example Input 4

```
     Prompt: Proprietary Scheme:
 User Input: ZZTOP
```

Example Output 4

```
Line 1: Invalid Input
```

Submissions must be in the following form:

- **xxxxxx** serves as a placeholder for your initials. (See Challenge Rules)
- C++: A single C++ source file named **xxxxxx_challenge4.cpp**, containing a main function that returns the integer zero on successful exit and non-zero in case of error.
- Java: A single Java source file named **xxxxxx_challenge4.java**, containing an appropriately named public class and public main method.
- Python3: a single python3 file named **xxxxxx_challenge4.py**
- C++, Java, and Python programs must compile/execute with no additional third-party libraries linked on build time.

Coding Challenge # 4 Rubric

- 2 points - The program correctly prompts and allows the user to enter a Base-16P value to be encoded.
- 2 points - The program correctly handles invalid input.
- 10 points - Output the correct ASCII characters to the console.

Coding Challenge # 5 - Marky-Markup and the Funky-Files

Points = 20

Narrative

The year is 1993 in an alternate universe where Marky Mark and The Funky Bunch were elected Benevolent Dictators for Life. As a token of your respect to their leadership, your team is wanting to develop a way to display entwinepage on the externet (things are a little off here) and decided to implement the latest and greatest in modern entwine design technologies. These files (called markup files, naturally) allow developers to redact sensitive information while also doing some conditional formatting. So, get your team to work while giving off those good vibrations!

Instructions

Write a program that will:

- Read in a file named ***input_file.marky*** located in the same directory as the program with no nested subfolders.
- Output to the console `File Doesn’t Exist` if the ***`input_file.marky`*** doesn’t exist.
- Create a new file in the same directory as the original file with the following name ***output_file.txt***.
- The output file’s contents should reflect the following commands documented below found in the ***input_file.marky***.

Command	**Documentation**
`{UPPER_X}`	Uppercase the next `X` characters. Characters that are already uppercase stay uppercase. This does not affect non-alpha characters which should stay the same before and after.
`{LOWER_X}`	Lowercase the next `X` characters. Characters that are already lowercase stay lowercase. This does not affect non-alpha characters which should stay the same before and after.
`{REDACT_X}`	The next `X` characters are replaced with a `#` symbol. This affects ALL characters.
`{REVERSE_X}`	Reverse the order of the next `X` characters. This affects ALL characters.
`{SPACE_X}`	Insert `X` “space” characters before the following character.
`{THEN}`	This sequences two commands together. The command on the left of the `{THEN}` is performed before the command on the right. There will be no more than two commands within a combined command when using `{THEN}`.

- No single command will target characters on multiple lines.
- See the example file contents to determine the structure of each file's contents before and after the program runs.
- The highlighted areas in the example file contents are to help with demonstrating the process of the program and are not in the actual files.
- The **`input_file.marky`** that the judges use to score can have any number of lines but will follow the format of the example file contents below.

- The program should not repeat and it should execute without error.

Example `input_file.marky` contents

```
{UPPER_5}Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. {LOWER_1}Ut
enim ad minim veniam, quis nostrud exercitation llamco laboris nisi ut
aliquip ex ea {REDACT_17}commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. {SPACE_4}Excepteur sint occaecat cupidatat non proident,
{LOWER_4}{THEN}{REVERSE_13}SUNT in CUlpa qui officia deserunt mollit anim id est laborum.
```

Example `output_file.txt` contents

```
LOREM ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. ut
enim ad minim veniam, quis nostrud exercitation llamco laboris nisi ut
aliquip ex ea #################. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur.     Excepteur sint occaecat cupidatat non proident,
aplUC ni tnus qui officia deserunt mollit anim id est laborum.
```

Example for output to the console when input_file.marky doesn't exist

```
File Doesn’t Exist
```

Submissions must be in the following form:

- **xxxxxx** serves as a placeholder for your initials. (See Challenge Rules)
- C++: A single C++ source file named **xxxxxx_challenge5.cpp**, containing a main function that returns the integer zero on successful exit and non-zero in case of error.
- Java: A single Java source file named **xxxxxx_challenge5.java**, containing an appropriately named public class and public main method.
- Python3: a single python3 file named **xxxxxx_challenge5.py**
- C++, Java, and Python programs must compile/execute with no additional third-party libraries linked on build time.

Coding Challenge # 5 Rubric

- 2 points - The program correctly handles for when the ***input_file.marky*** doesn't exist.
- 2 points - The program creates the file ***output_file.txt*** in the same directory as the program.
- 16 points - The program correctly implements the commands to generate the expected content in the ***output_file.txt*** file. Partial points may be awarded.

Coding Challenge # 6 - Welcome To The Desert Of The Real

Points = 20

Narrative

“This NeoCoin wallet has to be around here somewhere…Found it!” We’ll need to recover the seed phrase somehow before the machines can crack it. We know the Architect was the original owner of the wallet and used a specific algorithm to create the seed phrase. We just have to retrace their steps. We also happen to know something about the first word in their seed phrase. On a previous run one of the other ships was even able to visit The Source and nab the files used to generate the passphrase. Free your mind and the crypto-riches will follow!

Instructions

Write a program that will search through several files in the same directory to generate a phrase using the following requirements:

- Prompt the user to enter a starting point.
  - See example inputs in this challenge to determine the format of prompts and user input.
- Accept user input for a starting point on the next line in the console.
  - The user input format:

    `<START_ID>:<START_FILENAME>:<PHRASE_LENGTH>`

  - An example of the prompt and user input:

```
               Example Console
    Prompt:    Starting Point:
User Input:    0001:example_file_43.txt:1000
```

- Use the provided input to find the starting file in the same directory as the program with no nested subfolders.
- Output the starting file’s ID line found on the first line of the file.
  - This should be the only file ID line that will be printed by the program.

First line of example_file.txt

```
Line 1   ---File ID 0123---
```

- Use the START_ID from the provided input to find the first word in the phrase in the starting file.
  - All files will have text on some varying number of lines in the following format:

    `<ID>:<WORD>:<NEXT_FILENAME>:<NEXT_ID>`

    - An example of two lines that can be found in a file in the above format:

Snippet of example_file.txt contents

```
0001:Eli:example_file.txt:1010
1458:Smaug:example_file_2.txt:9456
```

- Go to the **<NEXT_FILENAME>** and find the line with the **<NEXT_ID>**.
    - All files will be in the same directory as the program with no nested subfolders.
    - Repeat this process **<PHRASE_LENGTH>** number of times storing each word in the order it is encountered.
- Output the phrase to the console with a space separating each word.
    - See example input and output for how the program should behave.
- The program should not repeat and it should execute without error.

Examples of files and their contents:

nebuchadnezzar.txt contents

```
---File ID 0123---
3333:has:mega_city.txt:0932
4520:has:mega_city.txt:1999
1337:Neo:zion.txt:7821
6727:white:mega_city.txt:9999
1532:Bane:nebuchadnezzar.txt:1337
2373:Agent:mega_city.txt:1336
```

zion.txt contents

```
---File ID 1234---
4598:wake:mega_city.txt:1234
7821:the:zion.txt:0101
0101:Matrix:nebuchadnezzar.txt:3333
9012:TrainMan:mega_city.txt:9999
1023:low:zion.txt:4598
4444:target:mega_city.txt:4444
7331:Neo:mega_city.txt:1001
```

mega_city.txt contents

```
---File ID 4597–
1336:Smith:nebuchadnezzar.txt:4520
0932:you:zion.txt:9012
1337:Trinity:mega_city.txt:9999
1234:up:nebuchadnezzar.txt:1337
9999:rabbit:nebuchadnezzar.txt:6727
6713:danger:nebuchadnezzar.txt:2373
1999:Morpheus:nebuchadnezzar.txt:2373
1001:wake:mega_city.txt:1001
4444:world:nebuchadnezzar.txt:1532
```

Example Input 1

```
     Prompt: Starting Point:
 User Input: 6727:nebuchadnezzar.txt:2
```

Example Output 1

```
     Line 1: ---File ID 0123---
     Line 2: white rabbit
```

Example Input 2

```
     Prompt: Starting Point:
 User Input: 4598:zion.txt:7
```

Example Output 2

```
     Line 1: ---File ID 1234---
     Line 2: wake up Neo the Matrix has you
```

Example Input 3

```
     Prompt: Starting Point:
 User Input: 6713:mega_city.txt:5
```

Example Output 3

```
     Line 1: ---File ID 4597---
     Line 2: danger Agent Smith has Morpheus
```

Submissions must be in the following form:

- **xxxxxx** serves as a placeholder for your initials. (See Challenge Rules)
- C++: A single C++ source file named **xxxxxx_challenge6.cpp**, containing a main function that returns the integer zero on successful exit and non-zero in case of error.
- Java: A single Java source file named **xxxxxx_challenge6.java**, containing an appropriately named public class and public main method.
- Python3: a single python3 file named **xxxxxx_challenge6.py**
- C++, Java, and Python programs must compile/execute with no additional third-party libraries linked on build time.

Coding Challenge # 6 Rubric

- 2 points - The program correctly prompts and allows the user to enter input required for the starting word.
- 4 points - Output the starting File ID line.
- 14 points - Output the correct phrase to the console.

Coding Challenge # 7 - Lord of the CipheRings

Points = 28

Narrative

The tech company your team works for has decided to make things more interesting at the office. Your team finds themselves at a table with one of the bosses claiming to be a "Dungeon Master" (That has something to do with Fortnite, right?). Anyway, after hours of creating character backstories, rolling dice, and the occasional "YAHTZEE!", the Dungeon Master presents your band of techfits with a puzzle. The Dungeon Master informs your team that to enter Moria you have to solve a cipher puzzle. You have a feeling that speaking the word "friend" is not enough to enter. With the information provided below, you and your team get to work in making heads or tails of the puzzle.

Instructions

Write a program that will decipher a message using the following information:

- Prompt the user to enter an encryption key.
  - See example inputs in this challenge to determine the format of prompts and user input.
- Accept an encryption key as user input on the next line in the console.
  - The encryption key format is explained later.
  - See below for the prompt and input format.

```
    Prompt: Enter Encryption Key:
User Input: r1_cc3-r2_c14-r3_cc2-r4_cc6-r5_c88
```

- Read in an encrypted message from the ***nothing_to_see_here.txt*** file in the same directory as the program.
  - The number of rows (lines) will be equal to the number of characters in a row (N x N).
  - See the below ***nothing_to_see_here.txt*** file contents for file structure.

nothing_to_see_here.txt example contents

```
GXTTHAMSNN
CYHHLYEXBE
TAGBFJOVOJ
SBRPPOYDUU
VNQPDBAZRT
GQVEWZHJFQ
RCGKZLGYSI
JKXIUAISNC
NNXHRJIQES
MMJKMXUGWO
```

- The steps for the encryption algorithm used to encrypt the above message is as follows.
    - An example message is provided to demonstrate the encryption process.

Original Message

F	A	R	O	V	E	R	T	H	E
M	I	S	T	Y	M	N	T	S	C
O	L	D	T	O	D	U	N	G	E
O	N	S	D	E	E	P	A	N	D
C	A	V	E	R	N	S	O	L	D
W	E	M	U	S	T	A	W	A	Y
E	R	E	B	R	E	A	K	O	F
D	A	Y	T	O	S	E	E	K	T
H	E	P	A	L	E	E	N	C	H
A	N	T	E	D	G	O	L	D	D

- Step 1: Rotate the characters in each ring based on the encryption key. An example of an encryption key is found below.

`r1_cc3-r2_c14-r3_cc2-r4_cc6-r5_c88`

- Encryption Key Format Guidelines
    - The encryption key will have rings and their associated rotation values.
    - Each ring will have exactly one rotation instruction ranging from 0 to 999 rotary-shift(s) in a clockwise or counterclockwise direction.
    - `r<N>` represents the ring number.
    - `cc<N>` means to rotate the ring Counter-Clockwise <N> positions.
    - `c<N>` means rotate the ring Clockwise <N> positions.
    - The ring Number and rotation directions are separated by an underscore (_).
    - The ring instruction groups are separated by a hyphen (**-**).
    - An example of an encryption key that corresponds to the **`nothing_to_see_here.txt`** and **`i_looked_anyway.txt`** example files is found on the next page.
        - It has been colored to match the corresponding rings in the file.

`r1_cc3-r2_c14-r3_cc2-r4_cc6-r5_c88`

Rotated Message

O	V	E	R	T	H	E	C	E	D
R	C	N	E	E	L	A	P	E	D
A	K	O	D	U	N	A	O	A	Y
F	O	T	A	E	R	B	W	R	F
M	A	D	A	R	N	U	K	E	T
O	L	S	S	S	T	E	E	A	H
O	N	V	P	E	E	D	E	N	D
C	G	M	E	Y	T	O	S	L	D
W	S	N	T	M	Y	T	S	I	L
E	D	H	A	N	T	E	D	G	O

- Step 2: Shift each character in a column by the column number with zero as the leftmost column.
    - Use the position of the letter in the alphabet with “A” being at position 0, “B” at position 1, “C” at position 2 … and “Z” at position 25.
    - For example, an “A” on column 2 will shift to “C” and a “Z” in column 9 will shift to “I”.

Column Shifted Message

0	1	2	3	4	5	6	7	8	9
O	W	G	U	X	M	K	J	M	M
R	D	P	H	I	Q	G	W	M	M
A	L	Q	G	Y	S	G	V	I	H
F	P	V	D	I	W	H	D	Z	O
M	B	F	D	V	S	A	R	M	C
O	M	U	V	W	Y	K	L	I	Q
O	O	X	S	I	J	J	L	V	M
C	H	O	H	C	Y	U	Z	T	M
W	T	P	W	Q	D	Z	Z	Q	U
E	E	J	D	R	Y	K	K	O	X

- Step 3: Shift each character in a row by the row number with zero being the topmost row.
    - Use the position of the letter in the alphabet with “A” being at position 0, “B” at position 1, “C” at position 2 … and “Z” at position 25.
    - For example, an “A” on row 2 will shift to “C” and a “Z” in row 9 will shift to “I”.

Row Shifted Message

0	O	W	G	U	X	M	K	J	M	M
1	S	E	Q	I	J	R	H	X	N	N
2	C	N	S	I	A	U	I	X	K	J
3	I	S	Y	G	L	Z	K	G	C	R
4	Q	F	J	H	Z	W	E	V	Q	G
5	T	R	Z	A	B	D	P	Q	N	V
6	U	U	D	Y	O	P	P	R	B	S
7	J	O	V	O	J	F	B	G	A	T
8	E	B	X	E	Y	L	H	H	Y	C
9	N	N	S	M	A	H	T	T	X	G

- Step 4: Flip all columns vertically across the horizontal axis.

Vertically Flipped Message

0	1	2	3	4	5	6	7	8	9
N	N	S	M	A	H	T	T	X	G
E	B	X	E	Y	L	H	H	Y	C
J	O	V	O	J	F	B	G	A	T
U	U	D	Y	O	P	P	R	B	S
T	R	Z	A	B	D	P	Q	N	V
Q	F	J	H	Z	W	E	V	Q	G
I	S	Y	G	L	Z	K	G	C	R
C	N	S	I	A	U	I	X	K	J
S	E	Q	I	J	R	H	X	N	N
O	W	G	U	X	M	K	J	M	M

- Step 5: Flip all rows horizontally across the vertical axis.

Horizontally Flipped Message (Encrypted)

0	G	X	T	T	H	A	M	S	N	N
1	C	Y	H	H	L	Y	E	X	B	E
2	T	A	G	B	F	J	O	V	O	J
3	S	B	R	P	P	O	Y	D	U	U
4	V	N	Q	P	D	B	A	Z	R	T
5	G	Q	V	E	W	Z	H	J	F	Q
6	R	C	G	K	Z	L	G	Y	S	I
7	J	K	X	I	U	A	I	S	N	C
8	N	N	X	H	R	J	I	Q	E	S
9	M	M	J	K	M	X	U	G	W	O

- Output the original, decrypted message to a file named ***i_looked_anyway.txt*** in the same directory as the program.
    - See the below ***i_looked_anyway.txt*** file contents for file structure.
- The program should not repeat and it should execute without error.

Example of Before and After Program Execution

nothing_to_see_here.txt contents before and after running program

```
GXTTHAMSNN
CYHHLYEXBE
TAGBFJOVOJ
SBRPPOYDUU
VNQPDBAZRT
GQVEWZHJFQ
RCGKZLGYSI
JKXIUAISNC
NNXHRJIQES
MMJKMXUGWO
```

Example Input 1

```
    Prompt: Enter Encryption Key:
User Input: r1_cc3-r2_c14-r3_cc2-r4_cc6-r5_c88
```

`i_looked_anyway.txt` contents after running program

```
Line 1   FAROVERTHE
Line 2   MISTYMTNSC
Line 3   OLDTODUNGE
Line 4   ONSDEEPAND
Line 5   CAVERNSOLD
Line 6   WEMUSTAWAY
Line 7   EREBREAKOF
Line 8   DAYTOSEEKT
Line 9   HEPALEENCH
Line 10  ANTEDGOLDD
```

Submissions must be in the following form:

- **xxxxxx** serves as a placeholder for your initials. (See Challenge Rules)
- C++: A single C++ source file named **xxxxxx_challenge7.cpp**, containing a main function that returns the integer zero on successful exit and non-zero in case of error.
- Java: A single Java source file named **xxxxxx_challenge7.java**, containing an appropriately named public class and public main method.
- Python3: a single python3 file named **xxxxxx_challenge7.py**
- C++, Java, and Python programs must compile/execute with no additional third-party libraries linked on build time.

Coding Challenge # 7 Rubric

- 2 points - The program correctly prompts and allows the user to enter an encryption key.
- 2 points - Create the file **`i_looked_anyway.txt`**.
- 24 points - The **`i_looked_anyway.txt`** file contents correctly reflect the decryption algorithm. Partial points may be awarded for the incremental implementation of the phases required to decrypt the message if the complete process is not implemented.