

Contestant Number: \_\_\_\_\_

Time: \_\_\_\_\_

Rank: \_\_\_\_\_



# PYTHON PROGRAMMING-PILOT (355)

## REGIONAL – 2021

### PRODUCTION PORTION:

Program 1: Low-Level Encryption \_\_\_\_\_ (310 points)

*TOTAL POINTS* \_\_\_\_\_ (310 points)

## Test Time: 90 minutes

**GENERAL GUIDELINES:**

*Failure to adhere to any of the following rules will result in disqualification:*

1. Contestant must hand in this test booklet and all printouts if any. Failure to do so will result in disqualification.
2. No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests (handwritten, photocopied, or keyed) are allowed in the testing area.
3. Electronic devices will be monitored according to ACT standards.

*You will have ninety (90) minutes to complete your work.*

Your name and/or school name should *not* appear on work you submit for scoring.

1. Create a folder on the flash drive provided using your contestant number as the name of the folder.
2. Copy your entire solution/project into this folder.
3. Submit your entire solution/project so that the graders may open your project to review the source code.
4. Ensure that the files required to run your program are present and will execute on the flash drive provided.

\*Note that the flash drive letter may *not* be the same when the program is scored as it was when you created the program.

\*It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter.

The graders will *not* compile or alter your source code to correct for this.  
Submissions that do *not* contain source code will *not* be scored.

**Assumptions to make when taking this assessment:**

- The user will *not* enter invalid input.
- The input file will only contain valid input strings that only contain upper and lower case letters.

**Development Standards:**

- Your Code must use a consistent variable naming convention.
- All functions (if any) must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).
- If you create a class, then you must document the class and its methods.

## Low-Level Encryption and Decryption

As an entry-level programmer, you have been asked to create a set low level encryption and decryption algorithms. Each algorithm will be in the form of a function. You will also create a program that asks the user to enter a string input. This program will then encrypt the inputted string and print it to the screen. The program will then decrypt the string and print the original string back to the screen.

### Input:

The program will read a series of strings from an input file named “plainText.txt”. There will be only one string per line with no spaces.

### Encryption / Decryption:

The encryption / decryption will be done in two phases. The first phase will be a “Reverse Cypher”. An example of this would be the string “Zoo” will now be stored as “ooZ”. The second phase will use a “Ceasar Cypher”. This cypher will rotate the letters of the string by 10 positions. All upper case letters will need to remain upper case after encryption and lower case letter must remain lower case. An example would be the string “ooZ” will be encrypted to “yyJ”. You will notice that the letter “Z” when rotated 10 positions became the letter “J”. A couple of portions of the ASCII table has been provided to help you with this part.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122

The decryption will require you to reverse the two phases used in the encryption phase.

### Output:

Your program will need to write the encrypted strings to a file named “encryptedText.txt” and to the screen. After this is completed, your program will need to read the encrypted strings from “encryptedText.txt” and decrypt. Once they are decrypted, the strings should be written to the screen. When the program is run, the output should resemble the example below.

#### Sample Run

Encrypted:

yyJ

hyP

owkXdckV

owkXdcbSP

kmsbowKpYcodkdCnodsxE

Decrypted:

Zoo

Fox

LastName

FirstName

UnitedStatesOfAmerica

**Requirements:**

1. You must create an application called Encryption.
2. Your program will contain a function for encryption that will take in a string and return the encrypted string.
3. Your program will contain a function for Decryption that will take in a string and return the decrypted string.
4. Your contestant number must appear as a comment at the top of the main source code file.
5. The program will display the output like the example above.

### Solution and Project

The project is present on the flash drive \_\_\_\_\_ 15 points  
The project is named Encryption \_\_\_\_\_ 15 points

### Program Execution

The program runs from the USB flash drive \_\_\_\_\_ 15 points  
*If the program does not execute, then the remaining items in this section receive a score of zero.*

The program runs to completion with no error codes \_\_\_\_\_ 20 points  
There are the same number of encrypted strings as decrypted strings \_\_\_\_\_ 20 points

### Source Code Review

The source code is properly commented  
    A comment containing the contestant number is present \_\_\_\_\_ 10 points  
Functions and code sections are commented \_\_\_\_\_ 20 points  
There are two separate functions for encrypting and decrypting \_\_\_\_\_ 30 points  
Each function has code to reverse the string \_\_\_\_\_ 20 points  
  
Each function has to code to encrypt/decrypt each character in the string  
while maintaining the character's case (upper or lower) \_\_\_\_\_ 30 points  
  
Code is present to read the strings from the file \_\_\_\_\_ 30 points  
Code is present to write the strings to the file \_\_\_\_\_ 15 points  
  
All files are opened and closed properly \_\_\_\_\_ 30 points  
Code uses a consistent variable naming convention \_\_\_\_\_ 10 points

**Total Points = \_\_\_\_\_ / 310 points**