

Laboratory 10

Topics

1. Reading and writing files
2. Exception handling
3. Data processing

Discussion

- A. What happens if you try to read or write open a file that does not exist?
- B. What is the difference between reporting and handling an exception?
- C. What is the difference between sequential access and random access?

Exercises

Part 1 - File Processing

Assignment: for each of the following exercises, write a program in Python that meets the given requirements. Complete at least one exercise during the lab session, and the remaining ones at home.

10.1.1 Enola Gay. Write a program that reads the text file `input.txt`, and writes each line read in the file `output.txt` preceded by its line number inserted as a comment between characters `'/*'` and `'*/'`. If, for example, the `input.txt` file contains the text:

```
Enola Gay
is the bomber who, on August 6, 1945,
dropped the first atomic bomb on Hiroshima.
nicknamed Little Boy.
```

the generated `output.txt` file must contain the text:

```
/*1*/Enola Gay
/*2*/is the bomber who, on August 6, 1945,
/*3*/dropped the first atomic bomb on Hiroshima.
/*4*/nicknamed Little Boy.
```

[P7.2]

10.1.2 From the bottom. Write a program that reads all the lines in a text file `input.txt`, reverses their order, and writes them to another file `output.txt`. For example, if the `input.txt` file contains the text:

```
Mary had a little lamb
Its fleece was white as snow
And everywhere that Mary went
The lamb was sure to go.
```

the generated `output.txt` file must contain the text:

```
The lamb was sure to go.
And everywhere that Mary went
Its fleece was white as snow
Mary had a little lamb
```

[P7.9]

10.1.3 Ring Search. Write a program that searches for a given word in the contents of a group of files. The program must ask the user for input:

- I. A list of **file names** on a single line, separated by commas;
- II. A word to search.

File names must be stored in a list, while the word to be searched must be stored in a variable. The program must display all lines that contain the word, even as a substring of other words, without distinguishing between uppercase and lowercase letters. Each displayed line must be preceded by the name of the file in which it is located. For example, if the word to be searched is `'ring'`, and the list contains the files:

```
book.txt, address.txt, homework.py
```

then the program, having processed the contents of those files, should display the rows where the word to be searched is found with the following format:

```
book.txt: There is only one Lord of the Ring, only one who can bend it to
his will
book.txt: The ring has awoken; it's heard its master's call.
address.txt: Kris Kringle, North Pole
address.txt: Homer Simpson, Springfield
homework.py: string = "text"
```

[P7.6]

10.1.4 Hotel. The administrative manager of a hotel records sales in a text file. Each line contains the following 4 information, separated by semicolon characters (`' ; '`):

- I. the client's name;
- II. the service sold;
- III. the amount paid;
- IV. the date of the event.

The services sold may be, for example, a dinner, a conference, or lodging. The proper format for this file is for it to contain 4 fields per line, and for the amount paid to contain values of type `float`.

Write a program that reads this text file, and displays the total amount for each type of service, reporting an error if the file does not exist or if its format is incorrect. [P7.29]

10.1.5 Second possibility. Write a program that asks the user to enter a series of values of type `float`. When the user enters a value that is not of the correct type, the program must give the user a second chance to enter the value, and after two chances it must stop reading the input, terminating the program immediately. Data entry continues until the user enters an empty string (`' '`) as input. Sum all correctly specified values and display their sum when the user has finished entering data. Use exception handling to detect improper input. [P7.13]

Part 2 - Data processing: matching and encryption

Assignment: for each of the following exercises, write a program in Python that meets the given requirements. Complete at least one exercise during the lab session, and the remaining ones at home.

10.2.1 Random monoalphabetic cipher. Caesar's cipher is a monoalphabetic substitution cipher, in which each letter in the plaintext is replaced in the ciphertext by the letter that is a certain number of positions later in the alphabet. Because the number of positions used for translation is fixed, Caesar's cipher is too easy to hack. To make the ciphertext more difficult to decipher, an alternative strategy is to use a word as the key instead of a number. For example, if the word that is used as the key is FEATHER, first the duplicate letters are removed, yielding FEATHR, and then the remaining letters of the alphabet are added in the queue, in reverse order. This results in the following sequence.

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

Then, the letters are encrypted according to the following scheme.

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

Any other characters (e.g., spaces, digits, or punctuation marks) must remain unchanged.

Write a program that requests as input from the user the name of a text file to be encrypted or decrypted, the choice of which operation to perform (encryption or decryption), a keyword, and the name of a file in which to write the output of the processing. The program must store the keyword entered by the user in a variable, and use it to encrypt or decrypt the text file provided as input, then write the result of the processing to the indicated output file. [P7.20]

10.2.2 University transcript. Write a program that displays a list of examinations passed by a student, with their grades. A `classes.txt` file is available, which contains the codes for all courses delivered in the educational institution (a U.S. college), the contents of which will be similar to this one:

CSC1
 CSC2
 CSC46
 CSC151
 MTH121 ...

For each course, there is a file whose name corresponds to the course code, followed by `'.txt'`. This file lists the people who passed the course exam, showing for each the identification number (`'Student ID'`) and the grade obtained. For example, the file `CSC2.txt` might contain the text:

```
11234 A-
12547 B
16753 B+
21886 C ...
```

Write a program that prompts the user to enter a `'Student ID'` as input, and then displays the list of passed exams associated with it, reporting the grades obtained for each exam, respecting the following format.

Student ID: 16753

```
CSC2 B+
MTH121 C+
CHN1 A PHY50
A-
[P7.28]
```

10.2.3 Playfair cipher. It is possible to decipher a text by simple analysis of the frequency of occurrence of individual letters. One way to thwart this strategy is to jointly encrypt pairs of letters. The Playfair cipher is a simple scheme that applies this strategy. In this scheme, first a keyword is chosen, and the letters duplicated in it are eliminated. Then you place the keyword thus worked out in a 5×5 checkerboard, followed by the remaining letters of the English alphabet in order. Since there are only 25 squares and the English alphabet has 26 letters, 'I' and 'J' are considered indistinguishable. Check that in the source text the letters are even, if not, add a 'Z'. Using 'PLAYFAIR', transformed to 'PLAYFIR', as the keyword, the following pattern is obtained.

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

Starting from this table, to encrypt a pair of letters, e.g., 'AM,' you locate the portion of the table that has 'A' and 'M' at two ends:

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

The encoding of the pair 'AM' is done by finding the values at the two remaining ends of the portion of the table, in this case, 'FH'. If the two letters of the pair are on the same row or on the same column, such as, the pair 'GO', the encoding is done by swapping the letters with each other, obtaining, in this case, 'OG'. Decryption follows the same rules. Write a program that encrypts or decrypts a text file according to the Playfair cipher scheme. [P7.23]

10.2.4 Covalent bonds. Suppose a file contains the energies and bond lengths for covalent bonds, such as those illustrated by the table below.

Bond (single , double , or triple)	Bond energy (kJ/mol)	Bond length (nm)
C C	370	0.154
C C	680	0.13
C C	890	0.12
C H	435	0.11
C N	305	0.15
C O	360	0.14
C F	450	0.14
C Cl	340	0.18
O H	500	0.10
O O	220	0.15
O Si	375	0.16
N H	430	0.10
N O	250	0.12
F F	160	0.14
H H	435	0.074

The format of the file (denoted `bond_data.txt`) requires that each row in the table correspond to a row of text, and that, in each row, the three fields are separated by a space (' '), and does not provide a header. For example, a file containing the data presented in the table might contain the text:

```
C|C 370 0.154
C||C 680 0.13
C|||C 890 0.12 ...
```

Write a program that accepts data from one column as input and displays data from the other two columns in the file. If the input data has a match to multiple rows in the table, the program must return data from the other two columns to all rows that have a match to the input value. For example, a bond length of 0.12 must return both triple bond C|C and bond energy 890 kJ/mol, and single bond N|O and bond energy 250 kJ/mol. [P7.34]