Laboratory 6

Topics

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

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

6.1.1 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]

6.1.2 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]

6.1.3 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]

6.1.4 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]

6.1.5 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 – Advanced lists

6.2.1 Out-of-range. Write a program that contains an `out-of-range error`. What happens if you run the program? [R6.10]

6.2.2 Buffer. Write the pseudocode for an algorithm that, given a list of defined length, moves the elements "forward" one position (thus increasing their index by one unit), and moves the element at the last position to the first position. For example, the list `1 7 9 3 0 4`, after this operation, becomes the list: `4 1 7 9 3 0`. Write a program implementing the above described algorithm.

[R6.17]

6.2.3 Play dice. Write a program that generates a sequence of `20` random rolls of a dice, stores them into a list and displays the generated values marking the longest set of identical values with the following format:

```
1 2 5 5 3 1 2 4 3 (2 2 2 2) 3 6 5 5 6 3 1
```

If there are multiple sequences of identical values of maximum length, use the formatting shown to highlight the first in order of occurrence. [P6.16]

6.2.4 Table. Write the instructions for the execution of the following operation for a table in Python of m rows and n columns (dimensions are inserted by the user):

I. Initialize the table with values equal to zero (`0`);
II. Fill all the cells with values equal to one (`1`);
III. Fill the cells alternating `0` and `1` in a chess scheme;
IV. Fill with `0` only the cells of the upper row and of the lower one, leaving the rest of the table unchanged;
V. Fill with `1` only the cells of the right column and of the left column, leaving the rest of the table unchanged;
VI. Calculate and display the sum of all the elements.

After each operation, display the table. [R6.28]

6.2.5 Lists union. Write a function `merge(a, b)` that merges the two lists `a` and `b`, alternating one element of the first and one element of the second. If one list is shorter than the other, the items are alternated as long as possible, then the items left in the longer list are added, in order, to the bottom. Starting lists should not be changed. If, for example, the content of `a` is `1 4 9 16` and the content of `b` is `9 7 4 9 11`, the invocation of `merge(a, b)` return a new list composed of the following values: `1 9 4 7 9 4 16 9 11`. [P6.30]

6.2.6 Sorted lists union. Write a function `merge_sorted(a, b)` that merges two lists (which are assumed to be already sorted ascending), returning a new, ascending sorted list. Manage a current index for each list to keep track of portions already processed. Starting lists should not be modified. If, for example, the content of `a` is `1 4 9 16` and the content of `b` is `4 7 9 9 11`, the invocation of `merge_sorted(a, b)` return a new list composed of the following values: `1 4 4 7 9 9 9 11 16`. The method `sort()` and the function `sorted()` must not be used. [P6.31]