AGILE BUSINESS MODULES

Office +20 2508 8784 | Mobile +20 110 083 5896
www.agilebm.com | info@agilebm.com
1171/1172 Street 9, Mokattam, Cairo

Dear Candidate,

Greetings from us all at Agile Business Modules. We are excited that you have made it this far and wishing you all the best.

The next pages are intended to assess your coding skills. This gives us an insight about your ability to cope with our requirements comfortably without extra pressure. Hence, we highly appreciate your carefulness and integrity.

The following programs are required to be written in plain PHP code. No ready-made platforms are allowed. You should never quote piece of codes from online or offline references. All programs are supposed to run in a command prompt (php-cli mode).

The delivered code will be assessed against the following criteria:

- Submission time (the earlier, the better).
- Completeness (the more problems you solve the better).
- Actual program output (against desired output).
- Performance (program should complete in reasonable time).
- Neatness of code (indentation and naming conventions).
- Quality of code (comments, use of methods and classes).

Wish you all the best.

Human Resources Management Department

Agile Business Modules

1. Write a program that reads a list of strings from an input text file and prints them to an output text file, left aligned, one per line, in a triangular frame with leading and trailing spaces.

Example:

Input: (text file)

```
Break
this
challenge
you
are
the
hero
```

Output: (text file)

```
              *
             ***
            *****
           *******
          *********
         ***********
        * Break     *
       ** this      **
      *** challenge ***
     **** you       ****
    ***** are       *****
   ****** the       ******
  ******* hero      *******
***************************
```

2. Given two binary strings, A (of length 10) and B (of length 4), output 1 if B is an alternating subsequence of A and 0 otherwise.

Input:

A text file containing several lines of pairs of binary strings A and B separated by a single space.

Output:

A new text file with corresponding logical value of: 'B is an alternating substring of A'.

Example

Input: (text file)

1010110010 0100

1110111011 1001

1000101011 1011

(Note: red colors & underlines are for highlighting only)

Output: (text file)

1

0

1

3. An employee signs in/out through a biometric attendance device upon arrival to office and before departure. He might repeat his punch to make sure he did not miss it. Write a program that can match his attendance logs against his expected schedule. In case of multiple logs for the same attendance slot, use earliest log for arrival and latest for departure.

All times are in 24-hour time format. Blank schedule entries denote a weekend, you should not match logs on these days. Attendance logs and schedule are always on the same day.

Example

Inputs:

I- Schedule File: a text file with schedule values separated by comma "," in the following order: date, expected arrival time, expected departure time. Lines with no arrival/departure times are weekends.

2020-09-27, 09:00, 17:00

2020-09-28, 09:00, 17:00

2020-09-29, 09:00, 17:00

2020-09-30, 09:00, 17:00

2020-10-01, 09:00, 17:00

2020-10-02

2020-10-03

2020-10-04, 09:00, 17:00

II- Attendance Logs File: a text file with actual attendance times. Each line contains a single attendance record date (YYYY-MM-DD) and time (24-hour format)

2020-09-27 08:56

2020-09-27 17:14

2020-09-28 09:01

2020-09-28 09:03

2020-09-28 17:56

2020-09-29 09:14

2020-09-29 17:14

2020-09-29 18:02

2020-09-30 08:56

2020-09-30 08:57

2020-09-30 18:01

2020-10-02 12:02

2020-10-04 09:08

Output:

The output is a text file with the following comma "," separated values:

- Date (from schedule file)
- Matching Times
  - arrival time (from logs file) or "n/a" if no matching record
  - departure time (from logs file) or "n/a" if no matching record
- Absence (in case no matching record)
- weekend (in case original schedule record has no corresponding expected in/out times)

2020-09-27, 08:56, 17:14

2020-09-28, 09:01, 17:56

2020-09-29, 09:14, 18:02

2020-09-30, 08:56, 18:01

2020-10-01, absence

2020-10-02, weekend

2020-10-03, weekend

2020-10-04, 09:08, n/a