Cairo University
Faculty of Computers & Artificial Intelligence
Theory of Computations

Assignment 1 (Regex)

Instructions

- The assignment is submitted in groups of **3-4 students** from the **same lab** or the **same TA**
- The Deadline for submission is **on Saturday 22/3 at 11:59 pm**
- Submission will be on Google Classroom. No late submission, or through e-mail submission is allowed.
- Please submit one compressed folder with the **.java files**. The folder name should follow this structure: **ID1_ID2_ID3_ID4_GROUP.zip**
- In case of **Cheating,** you will get a **negative grade** whether you give the code to someone, take the code from someone/internet, or even send it to someone for any reason.

Requirements

- Solve all the problems.
- For all problems write a **Java program** to solve them using **regular expression** by following the **expected input & output format for each problem**.
- The **Java program** will take **ONE input text file** containing the input values for each problem, the program must save the output of each problem in **ONE output text file** following the required format for each problem.

Input file:

Starts with the number of the problem, followed by the input values for the problem, then **"end"** to indicate the input for this problem is finished and to move to the next problem.

Output file:

Starts with the number of the problem, followed by the corresponding output values of the input for the problem, then **"x"** to indicate the output for this problem is finished and to move to the next problem.

Important Notes:

- All the problems may be tested with **different test cases**, so ensure to write regular expressions to cover and handle **all edge cases**.
- The use of standard string operation functions (such as length, contains, etc.) is not permitted. You **must** solve all problems **using only regular expressions**.

Q1: Solve the following problems using regex [120 points]

Problem#1 [10 points]

Write a regular expression to validate MAC addresses.
A valid MAC address:

- Must consist of 12 hexadecimal digits
- The digits are grouped into 6 pairs, separated by colons (:) or hyphens (-)
- Each digit must be a valid hexadecimal character (0-9, A-F, or a-f)
- The total length of the MAC address must be 17 characters *(including separators)* or 12 characters *(without separators).*

→ Sample:

Content in input file:

```
1
00:1A:2B:3C:4D:5E
00-1A-2B-3C-4D-5E
001A2B3C4D5E
00:1A:2B:3C:4D:5G
end
```

Content in output file:

```
1
valid
valid
valid
invalid
x
```

Problem#2 [10 points]

→ Given the alphabet is only (a,b), can be uppercase letters

Write a regex to match a string in which every **a** is followed by exactly one or three **b**'s

→ Sample:

Content in input file:

```
2
bb
bab
babab
abbabbb
end
```

Content in output file:

```
2
valid
valid
valid
invalid
x
```

Problem#3 [10 points]

Validate dates using a regular expression.
The dates should be in one of the following formats:

- YYYY/MM/DD (e.g., 2024/03/11)
- YYYY-MM-DD (e.g., 2024-03-11)
- DD/MM/YYYY (e.g., 11/03/2024)
- D/MM/YYYY (e.g., 1/03/2024)
- DD/M/YYYY (e.g., 11/3/2024)
- DD-MM-YYYY (e.g., 11-03-2024)
- D-MM-YYYY (e.g., 1-03-2024)
- DD-M-YYYY (e.g., 11-3-2024)

→ Sample:

Content in input file:

```
3
11/3/2024
1st-03-2024
1 10 123
end
```

Content in output file:

```
3
valid
invalid
invalid
x
```

Problem#4 [10 points]

Write a regular expression to validate IP addresses.
A valid IP address:

- Must contain numbers between 0 and 255
- There must be exactly three dots separating the four numbers

→ Sample:

Content in input file:

```
4
192.168.1.1
192.168.1.256
end
```

Content in output file:

```
4
valid
invalid
x
```

Problem#5 [10 points]

Validate all C++ variables using a regular expression.

→ Sample:

Content in input file:

```
5
x
x1y2
_hello
1x
bad name
!name
end
```

Content in output file:

```
5
valid
valid
valid
invalid
invalid
invalid
x
```

Problem#6 [10 points]

→ Given the alphabet is only (a,b), can be uppercase letters

Write a regex to match a string that has an odd number of b's *(lower case or upper case)*

→ Sample:

Content in input file:

```
6
bbb
baBb
abAb
end
```

Content in output file:

```
6
valid
valid
invalid
x
```

Problem#7 [10 points]

→ Given the alphabet is only (a,b)

Using regex, extract strings with an **odd number of a's** and an **odd number of b's** from the user's input.

You should print**:**

- The number of matched substrings
- The matched substring
- Substring's start and end indices in square brackets *[Start Index, End Index]*

→ Sample:

Content in input file:

```
7
aabb
aabaaaaabaa
end
```

Content in output file:

```
7
**aabb**
number of matched substrings: 1
ab [1, 3]
-----------------------------------------
**aabaaaaabaa**
number of matched substrings: 2
aabaaaaa [0, 8]
ba [8, 10]
x
```

Problem#8 [10 points]

Using regex, extract words whose length is a multiple of **5** from an input string.

You should print

- The number of matched words
- The matched words
- Words' start and end indices in square brackets *[Start Index, End Index]*.

→ Sample:

Content in input file:

```
8
The match was a perfect reflection of their hard work and dedication
Ants move quickly over dry sand
end
```

Content in output file:

```
8
**The match was a perfect reflection of their hard work and dedication**
Number of matched words: 4
match [4, 9]
reflection [24, 34]
their [38, 43]
dedication [58, 68]
-----------------------------------------
**Ants move quickly over dry sand**
No word matches
x
```

Problem#9 [10 points]

Write a regular expression to parse log entries and extract:

- Timestamps in the format YYYY-MM-DD HH:MM:SS
- Log levels (e.g., INFO, ERROR, DEBUG, WARN)
- Log messages

You should print all the extracted data

→ Sample:

Content in input file:

```
9
[2023-10-15 14:30:45] [INFO] Application started
[2023-10-15 14:31:10] [ERROR] Failed to connect to database
[2023-10-15 14:32:00] [INFO] User logged in
end
```

Content in output file:

```
9
Timestamp: 2023-10-15 14:30:45, Level: INFO, Message: Application started
Timestamp: 2023-10-15 14:31:10, Level: ERROR, Message: Failed to connect to
database
Timestamp: 2023-10-15 14:32:00, Level: INFO, Message: User logged in
x
```

Problem#10 [10 points]

Write a regular expression to validate mathematical expressions

Mathematical expressions:

- May contain numbers, floating points, or variables
- May have multiple variables, numbers, and operators on both the left and right-hand side.
- The equation may be from any degree (e.g., 1^{st}, 2^{nd}, etc.)
- Consider operators (+-/*%^) only

- The equation does not contain parentheses

→ Sample:

Content in input file:

```
10
3x-3y=90
x^2+3x+2=6.5+y^-1
3x*-3y/z=90+n
=9
end
```

Content in output file:

```
10
valid
valid
valid
invalid
x
```

Problem#11 [10 points]

Write a regex to match a string with exactly **7 words**

→ Sample:

Content in input file:

```
11
The sky turned orange at sunset today
Creativity makes life more interesting
end
```

Content in output file:

```
11
valid
invalid
x
```

Problem#12 [10 points]

Write a regex to match a string where letters and digits alternate (e.g., a1b2c3)

→ Sample:

Content in input file:

```
12
1a8c0z
Aa1b6
end
```

Content in output file:

```
12
valid
invalid
x
```

Q2: Extract the sensitive information and replace it with asterisks (***) [80 points]

Given a story paragraph stored in test file, you should hide the sensitive information and data by replacing the sensitive information with ****.
The sensitive information is [email, Egyptian mobile phone number, address, bank account info (local account number, IBAN, Swift Code), and Egyptian national id number]

1. Instructions:

For each sensitive information, write a regex to replace this information with asterisk *****.

---

2. Examples and Clarifications for each regex:

1. **Email:**

   Must replace any email address such as:

   20210000@stud.fci-cu.edu.eg, user.name12@gmail.com, etc.

2. **Egyptian Mobile Phone Number:**

   Must replace any possible Egyptian phone number such as:

   01111100881, 01001100881, 01501100881, 01201100881, (+20)1111100881, etc.

3. **Address:**

   Must replace any possible address where address's structure:

   Building Number/Name, Street Name, Area/District/Neighborhood, City/Governorate, Postal Code (optional), Country (Egypt)

   such as:

   - Building 1, El Tahrir Street, Dokki, Giza, 12611, Egypt *(where 12611 is the postal code)*
   - El Nasr Street, Maadi, Cairo, Egypt
   - 10 El Hegaz Street, Near Roxy Square, Heliopolis, Cairo, Egypt
   - Apartment 7, 7th Floor, Building 77, 90th Street, New Cairo, Egypt
   - Talaat Harb Str., Downtown Cairo, 11511, Egypt *(where 11511 is the postal code)*

   Etc.

4. **Bank Account Info:**

   - **Local Account Number:**

     Must replace any possible bank account number, which consists of 17 consecutive digits such as:

     12345678901234567, 1234 5678 9012 34567

   - **IBAN:**

     Must replace any possible IBAN, it consists of 29 characters long with the country code at the start such as:

     EG38 0017 0005 0000 0012 3456 7890 1

     EG380017000500000012345678901

   - **Swift Code:**

     Must replace any possible Swift Code, it consists of either 8 or 11 characters such as:

     NBEGEGCX

5. **Egyptian National ID Number:**

Must replace any possible Egyptian national id, it consists of 14 digits:

Section	*Digits*	*Meaning*	*Example*
Century Digit	1 digit	2 = 1900s, 3 = 2000s	3
Year of Birth	2 digits	Last two digits of the year	01 → 2001
Month of Birth	2 digits	Month (01 - 12)	05
Day of Birth	2 digits	Day (01 - 31)	15
Governor Code	2 digits	Place of birth code (01-88)	14
Serial Number	5 digits	Unique personal number	12345

Follow the structure provided to write your regex.

---

3. Input example:

```
13
C:\filename.txt
end
```

---

4. Paragraph text file content example:

"My name is Shakespeare Michael. I am 59 years old, and after many years of struggling and wandering through time, I have finally succeeded in obtaining a National ID, a significant achievement that grants me citizenship in this strange world I find myself in. My ID number is 16404230198234, which marks my official entry into this era. In this new land, I have befriended several people who have assisted me in adapting to the modern world. One of my friends helped me create an email address: shekspear59@gmail.com. Although I was unfamiliar with such a concept, it took me some time to learn how to use it. Even more baffling, I had to pay 50 Euro to use this email service! I learned that I had to send this payment through something called a bank account. What a peculiar world this is! The people here do not trade money directly for goods and services. Instead, they use machines that accept money and store them in buildings called banks. For example, I used a bank called HBSC, which is a strange name, but apparently it holds the power to manage my finances. I now have an account number: 90987657483929124, which is used to define my wealth in this new world. I have come to appreciate many things about the future, and it all feels so thrilling. However, the most perplexing part is how people have become so reliant on machines. They no longer carry coins and paper currency; instead, everything is done electronically. If I need to contact anyone, I must use a phone number, which is yet another strange concept. My personal number is 01252225398, so feel free to call if you wish to speak to me. Life in this time is full of wonder, but it is also incredibly complex. Technology and systems are unlike anything I ever imagined during my days as a playwright, yet I am excited to explore more and adapt to this world of endless possibilities."

---

5. Expected Output:

The same text file *(filename.txt)* but with the updated paragraph, without the sensitive information:

"My name is Shakespeare Michael. I am 51 years old, and after many years of struggling and wandering through time, I have finally succeeded in obtaining a National ID, a significant achievement that grants me citizenship in this strange world I find myself in. My ID number is **********, which marks my official entry into this era. Cairo University Faculty of Computers & Artificial Intelligence Theory of Computations In this new land, I have befriended several people who have assisted me in adapting to the modern world. One of my friends helped me create an email address: **********. Although I was unfamiliar with such a concept, it took me some time to learn how to use it. Even more baffling, I had to pay 50 Euro to use this email service! I learned that I had to send this payment through something called a bank account. What a peculiar world this is! The people here do not trade money directly for goods and services. Instead, they use machines that accept money and store them in buildings called banks. For example, I used a bank called HBSC, which is a strange name, but apparently it holds the power to manage my finances. I now have an account number: **********, which is used to define my wealth in this new world. I have come to appreciate many things about the future, and it all feels so thrilling. However, the most perplexing part is how people have become so reliant on machines. They no longer carry coins and paper currency; instead, everything is done electronically. If I need to contact anyone, I must use a phone number, which is yet another strange concept. My personal number is **********, so feel free to call if you wish to speak to me. Life in this time is full of wonder, but it is also incredibly complex. Technology and systems are unlike anything I ever imagined during my days as a playwright, yet I am excited to explore more and adapt to this world of endless possibilities."
