Київський національний університет імені Тараса Шевченка

Механіко-математичний факультет

Реферат на тему:

**«Регулярні вирази»**

Виконав:

студент 2 курсу, 1-ша групи комп’ютерної математики

Чайчено Олег

Київ 2022

**ЗМІСТ**

Вступ

завдання 1).

завдання 2).

завдання 3).

завдання 4).

Висновки

**Вступ**

Регулярні вирази (Regular expressions або, коротко, regex - регулярки) - це поки що непопулярна і недооцінена тема в сучасному C++. Але в той же час розумне використання регулярних виразів може позбавити вас від написання безлічі рядків коду. Якщо у вас вже є якийсь досвід роботи в промисловості, але ви не вмієте використовувати регулярні висловлювання - ви розбазарюєте 20-30% власної продуктивності.

Регулярні вирази справді підходять для складних завдань, де рукописний код синтаксичного аналізу у будь-якому разі буде настільки ж повільним; і для надзвичайно простих завдань, коли зручність читання і надійність регулярних виразів переважують їх витрати на продуктивність.

Далі ми виконнаємо 4 завдання де нам потрібно буде знайти посилання(https),дати,числа і знову дати

Перші 4 завдачі ми виконуємо за допомогою string.h

Наступні за допомогою string

І останні 4 за допомогою regex.

**УСІ ТЕКСТОВІ ФАЙЛИ ЙДУТЬ З ЗАВДАННЯМИ**

**Завдання 1)**

**А)За допомогою string.h**

За допомогою string.h потрібно знайти посилання у тхт файлі.

while(fgets(line,255,file))

Дістаємо рядки з файлу поки вони є

if(strstr(line,word)!=NULL)

Якщо є входження word(“https”),то ми виводимо цей рядок.

**В)За допомогою string**

String h=”https”,позначаємо що будемо шукати

While(!file.eof()),за допомогою цього проходимо по рядкам файла

За допомогою if(file.find(h)!=-1),шукаємо чи є силка в рядку,якщо є то виводимо її

**C)За допомогою regex**

While(!file.eof()),за допомогою цього проходимо по рядкам файла

За допомогою if(file.find(h)!=-1),шукаємо чи є силка в рядку,якщо є то виводимо її.

regex regUrl("\\b((?:https?|ftp|file)://[-a-zA-Z0-9+&@#/%?=~\_|!:, .;]\*[-a-zA-Z0-9+&@#/%=~\_|])");

Регулярний вираз який допомогає нам знайти посиллання(https) у текстовому файлі

**regex\_search** пошук відповідань до регулярного виразу regUrl

Якщо такі знаходяться, то виводимо нашу строку

**Завдання 2)**

**Кожна інфомарція йде з нової строки(тобто дати та слово в одній стрічці бути не може),подається в таком виду:**

**Sdadadas**

**10.10.2010(або інші формати вказані у завданні)**

**Sdadasdas**

**11.11.2011**

**…**

**A) За допомогою string.h**

while(fgets(line,255,file)) ,за допомогою цього йдемо по файлу та дістаємо стрічку

Шукаємо в якому форматі йде дата(за домогою індексу точки(якщо line[2]='.' Та line[5]=’.’ Це один формат,якщо в других місцях,то інший),робимо перевірку вказаних форматів у завданні)

І потом за допомого if,else if,else записуємо у формат рік(число) назва місяця дата(число).Змінені дати виводимо у файлі output

**В)За допомогою string**

Також використовуємо char line,fgets для роботи з файлом.Знаходимо та змінюємо потрібне.Вивод у output

**C) За допомогою regex**

While(!file.eof()),за допомогою цього проходимо по рядкам файла

regex pattern("\\b\\d{4}[.]\\d{2}[.]\\d{2}\\b");//data with year(4 dig).month(2).day(2)

regex pattern1("\\b\\d{2}[.]\\d{2}[.]\\d{4}\\b");//data(2).month(2).year(4)

regex pattern2("\\b\\d{2}[/]\\d{2}[/]\\d{4}\\b");//data(2)/month(2)/year(2)

regex pattern3("\\b\\d{2}[.]\\d{2}[.]\\d{2}\\b");//data(2).month(2).year(2)

Це регулярні вирази для пошуку заданих форматів дат

d{2}[.]… значить 2 числа до крапки

d{4}[/]… значить 4 числа до слеша, далі зрозуміло

Використовуємо regex\_search для пошуку потрібних стрічок

If(regex\_search(line.c\_str(), result, pattern) || regex\_search(line.c\_str(), result, pattern1) || regex\_search(line.c\_str(), result, pattern2) || regex\_search(line.c\_str(), result, pattern3)))

Ця умова значить,що якщо там є хочаб один потрібний наш варіант,то ця стрічка підходить нам і ми починаємо робити щось з нею далі.

Змінюємо дати на потрібний формат за записуємо у файл output

Хочу помітити, що в данному випадку регулярні вирази зробили нашу роботу більш легкою та оптимальную,тому їх викорастаннє більш простіше та коротше ніж минулий код у А) та В).Код вийшов коротший на 50 стрічок

**Завдання 3)**

У нас є файл з числами типу

10

-1

1.5

2,4

Нам потрібно їх записита у вигляді

(10.0000)

(-1.0000)

(1.5000)

(2.4000)

**А)За допомогою string.h**

Фукнція bool isdigit(char\*) робить перевірку чи є строчка числом

while(fgets(line,255,file)) ,за допомогою цього йдемо по файлу та дістаємо стрічку

char c=’,’ для пошуку коми у числі

index=-1 використовуємо для пошуку коми,проходимо в циклі по кожному елменту рядка,якщо там є кома, то отримаємо її індекс у index

якщо коми нема,то index=-1

Далі якщо index!=-1 то у циклі змінюємо line[index]='.'

За допомогою atof змінюємо стрічку у інт і записуємо

у форматі “(%.4lf)” у файл testOleg12212

**В)За допомогою string**

Функція bool isdig(string s) робить перевірку чи є строчка числом

За допомогою цилку

While(!file.eof()),за допомогою цього проходимо по рядкам файла

Перевіряємо їх у isdig,якщо тру то шукаємо кому за допомогою char c = ',';

Та int index = s.find(c); чи є кома у рядку,якщо є то индекс повертає її номер і ми замінюємо її на крапку,якщо її нема то int index=-1; і тоді ми нічого не змінюємо.Далі в обох випадках використовуємо функцію stof(sring);

Отримане число записуємо у файл output

Якщо bool isdig(string s) повертає false, то це слово і ми його просто вносимо у файл output.

**C) За допомогою regex**

regex x("([\\d])"); допомогає зрозуміти число лі в строчці чи ні,

While(!file.eof()),за допомогою цього проходимо по рядкам файла

Використовуємо regex\_search для пошуку потрібних стрічок

Перевіряємо їх у isdig,якщо тру то шукаємо кому за допомогою char c = ',';

Та int index = s.find(c); чи є кома у рядку,якщо є то индекс повертає її номер і ми замінюємо її на крапку,якщо її нема то int index=-1; і тоді ми нічого не змінюємо.Далі в обох випадках використовуємо функцію stof(sring);

Отримане число записуємо у файл output

Хочу помітити що в цьому завданні REGEX допоміг мені не писати зайву функцію для перевірки,тобто код с REGEX вийшов більш простим та зрозумілим.

**Завдання 4)**

**Знайти дати та змінити їх на сьогоднішню**

**А)За допомогою string.h**

Використовую #include <time.h>

Для знаходження дати

while(fgets(line,255,file)) ,за допомогою цього йдемо по файлу та дістаємо стрічку

 if(line[2]=='.'||line[5]=='.'||line[2]=='/'||line[5]=='/'||line[4]=='.'||line[7]=='.')

Перевірка чи є це дата(всі формати другого завдання),якщо є то замість них записуємо ctime(&now)

Якщо не дата,то просто записуємо цю стрічку у файл

На виході отримуємо output

**B)За допомогою string**

Time.h для дати

int theDay = tim.tm\_mday; сьогоднішній день

int theMonth = tim.tm\_mon + 1; місяць

int theYear = tim.tm\_year + 1900; рік

Далі усе як у 2) завданні,знаходимо формати дат та замість них пишемо потрібну,якщо це не дата, то просто записуємо її у файл

**C) За допомогою regex**

time\_t tt = time(NULL);

localtime\_s(&tim, &tt);

int theDay = tim.tm\_mday;

int theMonth = tim.tm\_mon + 1;

int theYear = tim.tm\_year + 1900;

Для знаходження сьогоднішньої дати,

regex pattern("\\b\\d{4}[.]\\d{2}[.]\\d{2}\\b");//data with year(4 dig).month(2).day(2)

regex pattern1("\\b\\d{2}[.]\\d{2}[.]\\d{4}\\b");//data(2).month(2).year(4)

regex pattern2("\\b\\d{2}[/]\\d{2}[/]\\d{4}\\b");//data(2)/month(2)/year(2)

regex pattern3("\\b\\d{2}[.]\\d{2}[.]\\d{2}\\b");//data(2).month(2).year(2)

Коментарі пояснють принцип регулрки тут

While(!file.eof()),за допомогою цього проходимо по рядкам файла

Потім змінюємо усі дати на сьогодняшні,якщо це не дата,то просто записуємо.На виході отримуємо файл output

**Висновки**

Регулярні вирази дуже корисна річ,вони допомогають знайти потрібне у файлі набагто простіше та без зайвого коду.Їх правильне використаннє набагато полегчує роботу в великих проектах, де і так багато коду.

Особисто в моїх завданнях вони сильно допомгли.З регулярними виразами у мене виходило на 15-20% менше коду,що є добре.