ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- 🙡 🕮 🙣 -----



**BÁO CÁO PROJECT I**

***Báo cáo tuần 7***

Giảng viên: **Ngô Lam Trung**

Sinh viên thực hiện: Nguyễn Quang Long

Mã số sinh viên: 20194320

**Hà Nội, năm 2021**

Nội dung đã hoàn thành:

Từ câu 142 đến câu

**142. Is Subsequence?**

a. Đề bài

Cho 1 xâu s, xác định xem nó có phải xâu con của t hay không? (sử dụng regex)

b. Giải pháp

Thêm các phần tử vào pattern rồi dùng regexsearch

c. Code

bool solution(std::string t, std::string s) {

  std::string pattern = "";

  for (char ch : s) {

    pattern += ".\*["  + std::string{ch} + ']';

  }

  std::regex regex(pattern);

  return std::regex\_search(t, regex);

}

**143. Eye Rhyme**

a. Đề bài

2 xâu được gọi là eye rhym nếu có 3 ký tự cuối giống nhau. Cho 1 xâu gồm 2 xâu con phân cách bởi dấu \, kiểm tra xem 2 xâu có eye rhym không?

b. Giải pháp

Sử dụng regex

c. Code

bool solution(std::string pairOfLines) {

  std::regex pattern("^.\*(.{3})\t.\*(.{3})$");

  std::smatch match;

  std::regex\_search(pairOfLines, match, pattern);

  return match.str(1) == match.str(2);

}

**144. Program Translation**

a. Đề bài

Cho 1 đoạn code là 1 hàm, thêm vào các đối số của hàm đó các ký tự $ (nếu chưa có)

Ví dụ:

For

code =

"function add($n, m) {\t

return n + $m;\t

}"

and args = ["n", "m"], the output should be

solution(code, args) =

"function add($n, $m) {\t

return $n + $m;\t

}"

b. Giải pháp

Sử dụng regex

c. Code

std::string solution(std::string code,

                     std::vector<std::string> args) {

  std::string argumentVariants = "";

  for (size\_t i = 0; i < args.size(); i++) {

    argumentVariants += args[i];

    if (i < args.size() - 1) {

      argumentVariants += "|";

    }

  }

  std::regex regex("([^$])\\b(" + argumentVariants + ")\\b");

  std::string fmt = "$1$$$2";

  return std::regex\_replace(code, regex, fmt);

}

**145. Repetition Encryption**

a. Đề bài

Cho 1 đoạn văn, xác định xem trong đoạn văn có bao nhiêu cặp từ giống nhau liên tiếp?

Ví dụ

For letter = "Hi, hi Jane! I'm so. So glad to to finally be able to write - WRITE!! - to you!",  
the output should be  
solution(letter) = 4.

There are 4 pairs of consecutive identical words in the text. They are shown in different colors below:  
"**Hi, hi** Jane! I'm **so. So** glad **to to** finally be able to **write - WRITE**!! - to you!"

b. Giải pháp

Chuyển đoạn văn thành chữ cái in thường rồi sử dụng regex để tìm

c. Code

int solution(std::string letter) {

  std::transform(letter.begin(), letter.end(), letter.begin(), ::tolower);

  std::regex pattern("([a-z]+)[^a-z]+\\1\\b");

  return std::distance(

    std::sregex\_iterator(letter.begin(), letter.end(), pattern),

    std::sregex\_iterator()

  );

}

**146. Bug and Bugfixes**

a. Đề bài

Trong 1 số game role, die roll được xác định theo AdX±B với A và X là các số nguyên dương, A là số lần die được roll (bỏ qua nếu A bằng 1), X là số mặt của die và B là số trừ từ kết quả. Xác định xem số điểm lớn nhất có thể có?

Ví dụ:

rules = "Roll d6-3 and 4d4+3 to pick a weapon, and finish the boss with 3d7!"

đầu ra là  
solution(rules) = 43.

Vì

* d6-3cho biết một cuộn duy nhất của xúc xắc 6 mặt, với 3kết quả bị trừ. Vì vậy, số lượng tối đa có thể nhận được là như vậy 6 - 3 = 3.
* 4d4+3là viết tắt của các 4cuộn của một con xúc xắc 4 mặt, 3được thêm vào kết quả. Có thể nhận được 4 \* 4 + 3 = 19điểm.
* 3d7có nghĩa là 3cuộn của một con xúc xắc 7 mặt. Số lượng tối đa để có được với nó là 3 \* 7 = 21.

b. Giải pháp

Sử dụng regex để phân tách các số A X B

c. Code

int solution(std::string rules) {

  std::regex pattern("(\\d\*)d(\\d+)([+-]\\d+)\*");

  std::sregex\_iterator it(rules.cbegin(), rules.cend(), pattern);

  std::sregex\_iterator end;

  int res = 0;

  while (it != end) {

    int rolls = it->str(1).empty() ? 1 : std::stoi(it->str(1));

    int dieType = std::stoi(it->str(2));

    int formulaMax = rolls \* dieType;

    if (!it->str(3).empty()) {

      if (it->str(3)[0] == '-') {

        formulaMax -= std::stoi(it->str(3).substr(1));

      }

      else {

        formulaMax += std::stoi(it->str(3).substr(1));

      }

    }

    res += formulaMax;

    it++;

  }

  return res;

}

**147. LRC to SubRip**

a. Đề bài

Dịch tất cả các lời bài hát từ LRC sang SubRip

Ví dụ

For:

lrcLyrics = ["[00:12.00] Happy birthday dear coder,",

"[00:17.20] Happy birthday to you!"]

And songLength = "00:00:20", the output should be

solution(lrcLyrics, songLength) = [

"1",

"00:00:12,000 --> 00:00:17,200",

"Happy birthday dear coder,",

"",

"2",

"00:00:17,200 --> 00:00:20,000",

"Happy birthday to you!"

]

b. Giải pháp

Với mỗi phần tử trong mảng lrcLyric, tách riêng thành từng phần để xử lý rồi push vào mảng kết quả.

c. Code

function solution(lrcLyrics, songLength) {

    let times = lrcLyrics;

    times = times.map(function (val) {

        let output = val.substring(1, 9);

        if (parseInt(output.substring(0, 1)) >= 6) {

            output = "01:" + (output.substring(0, 1) - 6) + output.substring(1, 5) + "," + output.substring(6) + "0";

        } else {

            output = "00:" + output.substring(0, 5) + "," + output.substring(6) + "0";

        }

        return output;

    });

    times.push(songLength+",000");

    let output = [];

    for (let i=0; i<times.length-1; i++) {

        output.push("" + (i + 1));

        output.push(times[i] + " --> " + times[i+1]);

        output.push(lrcLyrics[i].substring(11));

        output.push("");

    }

    output.pop();

    return output;

}

**148. HTML Table**

a. Đề bài

Cho 1 xâu là đoạn mã HTML biểu diễn 1 bảng, hãy đưa ra nội dung tại hàng row, cột column

b. Giải pháp

Tách xâu theo hàng tạo thành 1 mảng rồi lấy phần tử thứ row. Sau đó lại tách theo cột rồi lấy phần tử thứ column. Cuối cùng lấy ra nội dung của cột đó

c. Code

function solution(table, row, column) {

    try {

        return table.split("<tr>")[row+1].split("<td>")[column+1].split("<")[0];

    } catch (\_) {

        return "No such cell";

    }

}

**149. Chess Notation**

a. Đề bài

Cho 1 xâu biểu diễn 1 bàn cờ theo thứ tự từng hàng từ trên xuống dưới và cột từ trái qua phải, mỗi hàng ngăn cách nhau bởi ký tự / . Các quân màu đen sẽ ký hiệu bằng chữ cái thường và quân trắng ký hiệu bằng chữ cái in hoa ('P' = pawn, 'N' = knight, 'B' = bishop, 'R' = rook, 'Q' = queen, 'K' = king) Các vị trí trống sẽ được nhóm vào và thay thế bằng 1 số (bằng số vị trí trống). Xoay bàn cờ 90 độ sang phải, hãy đưa ra xâu biểu diễn bàn cờ đó lúc này

A picture containing text

Description automatically generatedVí dụ:

Ban đầu:

"rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR"

Sau khi xoay:

"RP4pr/NP4pn/BP4pb/QP4pq/K2P2pk/BP4pb/NP4pn/RP4pr"

b. Giải pháp

Khi xoay bàn cờ sang phải 90 độ thì quân cờ tại vị trí hàng i cột j sẽ dịch chuyển đến vị trí hàng j cột 7-i. Đầu tiên, ta sẽ tách xâu ban đầu thành các hàng rồi xử lý từng hàng. Nếu duyệt đến vị trí không có quân cờ (biểu diễn bằng 1 số) thì ta sẽ thêm từng đó khoảng trắng vào mảng ở các vị trí tương ứng. Sau bước này, ta được 1 mảng 2 chiều gồm các quân cờ và khoảng trắng ở vị trí ban đầu. Sau đó, ta sẽ đưa vào xâu các phần tử ở các vị trí sau khi xoay 90 độ, đối với các khoảng trắng thì ta sẽ đếm số khoảng trắng liên tiếp rồi thay bằng số. Mỗi khi duyệt xong 1 hàng thì ta thêm 1 ký tự / (trừ hàng cuối cùng)

c. Code

std::string solution(std::string notation) {

    vector<vector<char>> board(8,vector<char>(8));

    int x = 0, y = 0;

    for (auto c : notation)

    {

        if (c == '/') {

            x++;

            y = 0;

        } else if ('1' <= c && c <= '8') {

            while (c > '0') {

                board[x][y] = ' ';

                c--;

                y++;

            }

        } else {

            board[x][y] = c;

            y++;

        }

    }

    string ans;

    for (int i = 0; i < 8; i++)

    {

        int space = 0;

        for (int j = 7; j >= 0; j--){

            if (board[j][i] != ' ') {

                if (space > 0) {

                    ans += '0' + space;

                    space = 0;

                }

                ans += board[j][i];

            } else {

                space++;

            }

            }

        if (space > 0)

            ans += '0' + space;

        if (i < 7)

            ans += '/';

    }

    return ans;

}