ĐẠI HỌC BÁCH KHOA HÀ NỘI

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

----- 🙡 🕮 🙣 -----



**BÁO CÁO PROJECT I**

***Báo cáo tuần 11***

Giảng viên: **Ngô Lam Trung**

Sinh viên thực hiện: Nguyễn Quang Long

Mã số sinh viên: 20194320

**Hà Nội, năm 2022**

Nội dung đã hoàn thành:

Từ câu 41 đến câu

41. Digit Degree

a. Đề bài

Gọi bậc của 1 số nguyên dương là số lần khi tính tổng của các chữ số của số đó đến khi chỉ còn 1 chữ số, ví dụ với n = 91 => 9+1 = 10 => 1+0 = 1 => bậc =2

Cho 1 số nguyên dương, tính bậc của nó

b. Giải pháp

Nếu input là số có 1 cs thì output là 0, ngược lại ta sẽ tính tổng các chữ số của nó rồi tính bậc của số này (sau khi tăng biến đếm thêm 1)

c. Code

int sumofNums(int n){

    int sum =0;

    while(n>0){

        sum+=n%10;

        n/=10;

    }

    return sum;

}

int solution(int n) {

    int count = 0;

    while(n>9){

        n=sumofNums(n);

        count++;

    }

    return count;

}

42. Bishop and Pawn

a. Đề bài

Cho vị trí của 1 con tượng và 1 con tốt trên bàn cờ vua. Kiểm tra xem con tượng có thể ăn được con tốt không

b. Giải pháp

Để con tượng có thể ăn được con tốt thì con tốt phải nằm trên đường chéo của con tượng, khi đó con tốt và con tượng là 2 đỉnh nằm trên 1 đường chéo của 1 hình vuông, tức khoảng cách hoành độ của 2 quân bằng khoảng cách tung độ của 2 quân.

c. Code

bool solution(string bishop, string pawn) {

    if(abs( (int)(pawn[0]-bishop[0]) ) == abs( (int) (pawn[1]-bishop[1]) ) ) return true;

    return false;

}

43. Is Beautiful String

a. Đề bài

1 xâu được gọi là beautiful nếu mỗi ký tự đứng sau trong bảng chữ cái không xuất hiện nhiều lần hơn ký tự đứng trước nó trong bảng chữ cái.

Ví dụ, xâu “aaabbbcce”, a xuất hiện 3 lần >= b xuất hiện 3 lần > c xuất hiện 2 lần > e 1 lần => xâu này là beautiful

b. Giải pháp

Sử dụng 1 mảng gồm 26 phần tử, mỗi phần tử lưu số lần xuất hiện của 1 ký tự, sau đó duyệt mảng, nếu phần tử đứng sau lớn hơn phần tử đứng trước thì trả về false. Ngược lại trả về true

c. Code

bool solution(string inputString) {

    int chaarracters[26]={0};

    for(char c: inputString){

        chaarracters[c-'a']++;

    }

    for(int i=0;i<25;i++){

        if(chaarracters[i]<chaarracters[i+1]) return false;

    }

    return true;

}

44. Find Email Domain

a. Đề bài

Cho 1 địa chỉ email hợp lệ, tìm tên miền của địa chỉ email đó.

b. Giải pháp

Tìm đến ký tự @ đầu tiên mà sau đó là chữ cái, tên miền sẽ là xâu con tính từ sau @ đến hết

c. Code

string findEmailDomain(string address) {

    int i=0;

     do{

         i = address.find('@',i+1);

    }

    while(!isalpha(address[i+1]));

    return address.substr(i+1);

}

45. Build Palindrome

a. Đề bài

Cho 1 xâu, tìm xâu palidrome ngắn nhất có thể tạo ra bằng cách thêm 1 số ký tự vào xâu đó.

b. Giải pháp

Thêm lần lượt từng ký tự từ đầu xâu vào cuối xâu cho đến khi tạo ra xâu palindrome

c. Code

bool palindrome(string st){

    for(int i = 0; i<st.length()/2;i++){

        if(st[i]!=st[st.length()-1-i]) return false;

    }

    return true;

}

string solution(string st) {

    int i=-1;

    while(1){

        string tmp = st;

        for(int j = i; j>=0;j--) tmp+=st[j];

        if(palindrome(tmp)) return tmp;

        i++;

    }

}

46. Elections Winners

a. Đề bài

Cho 1 mảng chỉ số phiếu hiện tại của những ứng cử viên và 1 số k là số người chưa bầu? Tìm xem có bao nhiêu người vẫn còn cơ hội thắng cử?

b. Giải pháp

Ta sẽ cộng lần lượt từng phần tử với k, nếu được số lớn hơn max thì count ++.

Có 1 số trường hợp đặc biệt: Nếu có nhiều người đang có cùng số phiếu lớn nhất và k = 0 thì return 0, nếu chỉ có 1 người có số phiếu lớn nhất và k = 0 thì return 1.

c. Code

int solution(vector<int> votes, int k) {

    int max = 0, count = 0, count2=0;

    for(int i: votes){

        if(max<i) max = i;

    }

    for(int i: votes){

        if(i==max) count2++;

        if(i+k>max) count++;

    }

    if(count2>1 && k==0) return 0;

    if(count2==1 && k==0) return 1;

    return count;

}

47. Is MAC48 Address?

a. Đề bài

Cho 1 xâu, kiểm tra xem xâu đó có phải địa chỉ MAC-48 không?

b. Giải pháp

-Nếu xâu có độ dài lớn hơn 17 thì không phải là địa chỉ MAC-48

-Duyệt qua từng ký tự trong xâu xem có nằm trong đoạn [A,F], [0,9] và các ký tự ở vị trí 2,5,8,… có phải “-” không?

c. Code

bool isMAC48Address(string inputString) {

    int i=0;

    if(inputString.length()>17) return false;

    do{

        if((inputString[i]<='9'&&inputString[i]>='0')||(inputString[i]<='F'&&inputString[i]>='A')) i++; else return false;

        if((inputString[i]<='9'&&inputString[i]>='0')||(inputString[i]<='F'&&inputString[i]>='A')) i++; else return false;

        if(inputString[i]!='-'&&i<17) return false;

        i++;

    }while(i<17);

    return true;

}

48.Is digit

a. Đề bài

Cho 1 ký tự, kiểm tra xem nó có phải số không

b. Code

bool solution(char symbol) {

    return isdigit(symbol);

}

49.Line Encoding

a. Đề bài

Cho 1 xâu, hãy biến đổi nó như sau:

Đầu tiên, các xâu được tách thành các xâu con sao mỗi xâu con là các ký tự giống nhau liên tiếp.

Sau đó mỗi xâu con sẽ được thay thế thành số ký tự + ký tự đó (trừ khi số ký tự = 1).

Cuối cùng hợp tất cả các xâu con thành 1 xâu

Ví dụ “aaabbc” => “3a2bc”

b. Giải pháp

Tách xâu đã cho thành các xâu con rồi thêm độ dài mỗi xâu con vào trước ký tự đầu tiên của 1 xâu để tạo thành các xâu còn mới. Cuối cùng cộng các xâu này lại với nhau

c. Code

string solution(string s) {

    vector<string> str;

    int i = 1;

    string tmp = ""; tmp+=s[0];

    while(i<=s.length()){

        if(s[i]==s[i-1]) tmp+=s[i];

        else {

            str.push\_back(tmp);

            tmp = ""; tmp+=s[i];

        }

        i++;

    }

    tmp = "";

    for(string st:str){

        if(st.length()!=1) tmp+= to\_string(st.length());

        tmp+=st[0];

    }

    return tmp;

}

50. Chess Knignt

Cho vị trí của 1 con mã trên bàn cờ vua, tính xem có bao nhiêu vị trí mà con mã có thể đi trong nước tiếp theo.

b. Giải pháp

Kiểm tra từng vị trí con mã có thể đi, nếu vị trí thỏa mãn nằm trong bàn cờ thì tăng biến đếm thêm 1.

c. Code

int solution(string cell) {

    int count =0;

    char x = cell[0];

    char y = cell[1];

    if(x-2>='a'&&y+1<='8') count++;

    if(x-1>='a'&&y+2<='8') count++;

    if(x+1<='h'&&y+2<='8') count++;

    if(x+2<='h'&&y+1<='8') count++;

    if(x+2<='h'&&y-1>='1') count++;

    if(x+1<='h'&&y-2>='1') count++;

    if(x-1>='a'&&y-2>='1') count++;

    if(x-2>='a'&&y-1>='1') count++;

    return count;

}

51. Delete Digit

a. Đề bài

Cho 1 số, nếu xóa 1 chữ số của số đó thì số lớn nhất có thể thu được là bao nhiêu

b. Giải pháp

Chuyển số đó thành xâu rồi xóa từng phần tử để tìm số lớn nhất

c. Code

int solution(int n) {

    int max = n/10;

    string n\_ = to\_string(n);

    for(int i =0; i<n\_.length();i++){

        string tmp = n\_;

        tmp.erase(tmp.begin()+i);

        if(max<stoi(tmp)) max = stoi(tmp);

    }

    return max;

}

52. Longest Word

a. Đề bài

Xác định từ dài nhất có trong 1 đoạn câu tiếng Anh

b. Giải pháp

Tách các từ có trong xâu đó thành các xâu con rồi lấy từ dài nhất

c. Code

string solution(string text) {

    vector<string> res;

    string tmp ="";

    for(int i = 0; i<text.length(); i++){

        if(isalpha(text[i])) tmp += text[i];

        else if(tmp!="") {

            res.push\_back(tmp);

            tmp = "";

        }

    }

    if(tmp!="") res.push\_back(tmp);

    int max = 0, pos;

    for(int i = 0; i<res.size();i++){

        if(max<res[i].size()) {

            max = res[i].size();

            pos = i;

        }

    }

    return res[pos];

}

53. Valid Time

a. Đề bài

Kiểm tra xem 1 xâu có phải là 1 xâu chỉ thời gian chính xác (trong đồng hồ 24h) không?

b. Giải pháp

Tách xâu ra giờ và phút, nếu giờ <24 và phút <60 thì thỏa mãn, ngược lại thì không.

c. Code

bool solution(string time) {

    int hour = (int)(time[0]-'0')\*10 + (int)(time[1]-'0');

    int minute = (int)(time[3]-'0')\*10 + (int)(time[4]-'0');

    if(hour<24&&minute<60) return true;

    return false;

}

54. Sum up Numbers

a. Đề bài

Cho 1 xâu ký tự (đoạn văn), tính tổng tất cả các số có trong xâu đó

b. Giải pháp

Tìm tất cả các số có trong xâu đó đưa vào mảng rồi tính tổng

c. Code

int solution(string inputString) {

    vector<int> res;

    string tmp = "";

    for(int i=0; i<inputString.size(); i++){

        if(isdigit(inputString[i])) tmp+=inputString[i];

        else if(tmp!="") {

            res.push\_back(stoi(tmp));

            tmp="";

        }

    }

    if(tmp!="") res.push\_back(stoi(tmp));

    int sum = 0;

    for(int i=0; i<res.size();i++){

        sum+=res[i];

    }

    return sum;

}

55. Different Squares

a. Đề bài

Cho 1 ma trận hình chữ nhật, tính số ma trận 2x2 khác nhau có trong ma trận đó

b. Giải pháp

Sử dụng CTDL set, lần lượt insert các ma trận 2x2 vào set, cuối cùng trả về kích thước của set

c. Code

int solution(vector<vector<int>> matrix) {

    set<vector<vector<int>>> set;

    for(int i =0; i<matrix.size()-1; i++){

        for(int j=0; j<matrix[0].size()-1; j++){

            set.insert({{matrix[i][j],matrix[i][j+1]},{matrix[i+1][j],matrix[i+1][j+1]}});

        }

    }

    return set.size();

}

56. Digits Product

a. Đề bài

Cho 1 số nguyên dương 0 <= product <= 600, tìm số nguyên dương nhỏ nhất có tích các chữ số bằng product

b. Giải pháp

Do product <=600 <9! Nên ta sẽ chia product lần lượt cho 2, 3, 4…, 9 (nếu chia hết). Nếu cuối cùng product = 1 thì product được tạo thành từ tích các số chia trên, khi đó ta sẽ tính số nhỏ nhất tạo ra từ các số chia đó, ngược lại nếu product >1 thì trả về -1

c. Code

int solution(int product) {

  if (product == 0) return 10;

  if (product == 1) return 1;

  int res = 0, div = 1;

  for (int i = 9; i > 1; --i) {

    while (product % i == 0) {

      product /= i;

      res += i \* div;

      div \*= 10;

    }

  }

  if (product > 1)

    res = -1;

  return res;

}

57. File Naming

a. Đề bài

Cho 1 mảng là các xâu thể hiện tên tệp. Các tên tệp không được giống nhau nên nếu tên nào trùng sẽ được thêm “(i)” với i là số lần trùng của nó.

b. Giải pháp

Duyệt lần lượt các xâu và thêm vào set, nếu xâu đã xuất hiện trong set thì thêm “(i)” vào sau xâu đó rồi thêm vào set, lặp lại đến khi không tìm thấy trong set nữa.

c. Code

std::vector<std::string> solution(std::vector<std::string> names) {

    vector<string> nm;

    set<string> s;

    for(string str: names) {

        int t{};

        string temp = str;

        while(s.find(temp)!=s.end())

            temp = str +"("+to\_string(++t)+")";

        s.insert(temp);

        nm.push\_back(temp);

    }

    return nm;

}

58. Message From Binary Code

a. Đề bài

Cho 1 đoạn mã nhị phân, hãy chuyển đoạn mã đó thành xâu ký tự ASCII biết mỗi byte là 1 ký tự

b. Giải pháp

Chuyển lần lượt từng 8 bit nhị phân liên tiếp thành cơ số 10 rồi chuyển thành mã ASCII, cuối cùng ghép các ký tự đó thành 1 xâu

c. Code

char bin\_to\_char(string byte){

    int res = 0;

    int k =0;

    for(int i =7; i>=0; i--){

        if(byte[i]=='1') res+=pow(2,k);

        k++;

    }

    return (char) res;

}

string solution(string code) {

    string res = "";

    int i =0;

    while(i<code.length()){

        string sub = code.substr(i,8);

        res+=bin\_to\_char(sub);

        i+=8;

    }

    return res;

}

59. Spiral Numbers

a. Đề bài

Hãy xây dựng ma trận nxn bằng cách thêm vào các số nguyên từ 1 đến n\*n theo hình xoắn ốc cùng chiều kim đồng hồ

b. Giải pháp

Thêm lần lượt các phần tử cho đến khi đến biên của ma trận thì đổi hướng theo thứ tự phải>xuống>trái>lên

c. Code

vector<vector<int>> solution(int n) {

    vector<vector<int>> res;

    res.resize(n,vector<int>(n));

    //  right = 0, down = 1, left = 2, up = 3

    int i=0, j=0, next=0;

    int infinity\_loop = 0;

    for(int k =1; k<=n\*n; k++){

        res[i][j] = k;

    if(j+1<n && !res[i][j+1] && next==0){

         j++;

         infinity\_loop = 0;

         continue;

         }

else if(i+1<n && !res[i+1][j] &&next==1){

         i++;

         infinity\_loop = 0;

         continue;

         }

else if(j-1>=0&& !res[i][j-1] &&next==2){

        j--;

        infinity\_loop = 0;

        continue;

         }

else if(i-1>=0&& !res[i-1][j] &&next==3){

        i--;

        infinity\_loop = 0;

        continue;

         }

        ++next%=4;

        k--;

        infinity\_loop ++;

        if(infinity\_loop>5) break;

    }

    return res;

}

60. Sudoku

a. Đề bài

Cho 1 ma trận 9x9, kiểm tra xem ma trận có phải sudoku không? (mỗi hàng, mỗi cột, mỗi ma trận con 3x3(không đè lên nhau) chỉ chứa các số 1 -> 9, các số không lặp lại)

b. Giải pháp

Duyệt qua từng hàng, cột và ma trận 3x3 để kiểm tra có phải sudoku không

c. Code

bool solution(std::vector<std::vector<int>> grid) {

    for (int i = 0; i < 9; i++) {

        int a = 0, b = 0, c = 0;

        for (int j = 0; j < 9; j++) {

            a ^= 1 << grid[i][j];

            b ^= 1 << grid[j][i];

            c ^= 1 << grid[i - i % 3 + j / 3][i % 3 \* 3 + j % 3];

        }

        if (a != 1022 || b != 1022 || c != 1022)

            return false;

    }

    return true;

}