**ĐỀ LUYỆN TẬP NGĂN XẾP + HÀNG ĐỢI KỲ HÈ 2024**

**BÀI 1. DSA\_P051. ĐẢO TỪ**

Cho một xâu ký tự str bao gồm nhiều từ trong xâu. Hãy đảo ngược từng từ trong xâu?

**Input**: Dòng đầu tiên đưa vào số lượng bộ test T. Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một dòng ghi lại nhiều từ trong xâu str.

**Output**: Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**: T, str thỏa mãn ràng buộc: 1≤T≤100; 2≤length(str)≤106.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  ABC DEF  123 456 | CBA FED  321 654 |

**BÀI 2. DSA\_P061. KIỂM TRA CÁC CẶP DẤU NGOẶC**

Cho biểu thức được tạo bởi các cặp “(, )”, “[, ]”, “{, }”.

Hãy cho biết các cặp “(, )”, “[, ]”, “{, }” viết đúng hay không đúng trong biểu thức

**Ví dụ:**

* exp = “[()]{}{[()()]()}” là biểu thức đúng (true)
* exp = “[(])” là sai (false).

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức exp.

**Output**:  Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**: T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤106.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  [()]{}{[()()]()}  [(]) | true  false |

**BÀI 3. DSA\_P052. ĐÁNH SỐ THỨ TỰ CẶP DẤU NGOẶC**

Cho một biểu thức đúng, tức là các dấu ngoặc đơn đều đầy đủ mở và đóng, đảm bảo đúng thứ tự. Hãy viết chương trình đánh số các cặp dấu ngoặc theo thứ tự xuất hiện, tính từ 1.

Ví dụ với biểu thức                                              (a + (b \*c) ) + (d/e)

ta có thứ tự của các cặp ‘(‘, ‘)’ là                         1  2  2  1  3  3

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T (không quá 100).
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một biểu thức số học được đưa vào trên một dòng, độ dài không quá 106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  (a + (b \*c) ) + (d/e)  ( ( () ) ( () ) ) | 1  2  2  1  3  3  1 2  3  3  2  4  5  5  4  1 |

**BÀI 4. DSA\_P091. THAY THẾ DẤU NGOẶC**

Cho một biểu thức toán học với các dấu ngoặc đơn, các toán hạng và các toán tử.

Với mỗi cặp dấu ngoặc đúng, hãy thay thế dấu mở ngoặc ( thành số 0 và dấu đóng ngoặc thành số 1.

Với các vị trí dấu ngoặc không đúng thì thay thế bằng số -1

Ví dụ biểu thức  ((a+b)- (c+d) thì sẽ được thay thế thành -10a+b1-0c+d1

**Input**

Dòng đầu ghi số bộ test.

Mỗi bộ test sẽ viết trên một dòng biểu thức có độ dài không quá 1000.

**Output**

Với mỗi bộ test, viết trên một dòng xâu ký tự sau khi đã thay thế dấu ngoặc.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  ((a)  (a))  (((abc))((d))))) | -10a1  0a1-1  000abc1100d111-1-1 |

**BÀI 5. DSA\_P220. NHẢY SANG PHẢI**

Cho N cây cột, trong đó cột thứ i có chiều cao là h[i].

Con ếch xuất phát từ một vị trí nào đó và chỉ được phép nhảy đến các cây cột gần nhất có chiều cao lớn hơn cột hiện tại.

Có Q truy vấn, mỗi truy vấn cho biết điểm xuất phát. Hãy tính số bước nhảy mà con ếch có thể thực hiện được.

**Input**

Dòng đầu ghi hai số N và Q (1 ≤ N, Q ≤ 105).

Dòng thứ hai ghi N số của dãy h[], các giá trị nguyên dương và không quá 109.

Q dòng tiếp theo, mỗi dòng ghi một vị trí xuất phát (chỉ số tính từ 1 đến N)

**Output**

Ghi ra Q dòng là số bước nhảy mà con ếch có thể thực hiện.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 5 1 3 4 2 5 1  2  3  4  5 | 3  2  1  1  0 |

**BÀI 6. DSA\_P053. ĐOẠN LIÊN TIẾP**

Cho dãy số A[] có N phần tử. Với mỗi vị trí thứ i trong dãy, hãy tính độ dài của đoạn liên tiếp tính từ i trở về phía trước mà các giá trị đều nhỏ hơn hoặc bằng A[i].

**Input:**Dòng đầu ghi số bộ test (không quá 10). Mỗi test có 2 dòng.

* Dòng đầu tiên gồm 1 số nguyên N (1 ≤ N ≤ 105).
* Dòng tiếp theo gồm N số nguyên A1, A2, …, AN (1 ≤ A[i] ≤ 106).

**Output**

* Với mỗi bộ test, in ra dãy kết quả trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  7  100 80 60 70 60 75 85 | 1 1 1 2 1 4 6 |

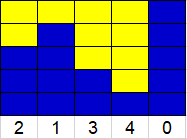
**BÀI 7. DSA\_P054. HÌNH CHỮ NHẬT ĐƠN SẮC**

Cho bảng có kích thước M\*N các ô vuông đơn vị.

Với mỗi dãy số A có N phần tử và các giá trị A[i] thỏa mãn nguyên dương, không vượt quá M, người ta tiến hành tô các ô trong bảng như sau:

* Tô từ trên xuống dưới trong từng cột
* Với cột thứ i, đầu tiên tô A[i] ô màu vàng, tiếp theo là M – A[i] ô màu xanh.

Ví dụ dưới đây là bảng 5\*5 với A[] = {2, 1, 3, 4, 0}.



Hãy tính diện tích hình chữ nhật lớn nhất thỏa mãn:

* Cạnh hình chữ nhật song song với cạnh của bảng
* Tất cả các ô trong hình chữ nhật đều cùng màu.

**Input**

Dòng đầu tiên ghi hai số M và N (0 < N,M <=106 ).

Dòng thứ 2 ghi N số của dãy A thỏa mãn ràng buộc đề bài.

**Output**

Ghi ra giá trị diện tích tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 9  1 3 4 4 5 4 4 3 1 | 21 |

**BÀI 8. DSA\_P207. TÍNH TOÁN GIÁ TRỊ BIỂU THỨC**

Biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng.

Ví dụ AB+CD-\* (tương ứng với biểu thức trung tố (A+B)\*(C-D).

Biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng.

Ví dụ + \* A B / C D tương ứng với biểu thức trung tố A \* B + C / D

Cho một biểu thức ở một trong hai dạng hậu tố hoặc tiền tố. Hãy tính giá trị biểu thức đó.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 50).

Mỗi test bắt đầu bởi số nguyên N (N ≤ 1000), là số lượng toán tử và toán hạng.

Dòng tiếp theo là biểu thức hậu tố hoặc tiền tố với N toán tử và toán hạng. Các toán hạng và toán tử cách nhau bởi một dấu cách.

Các toán hạng có thể có nhiều chữ số, có thể là số âm. Các toán tử chỉ bao gồm các phép toán cộng, trừ, nhân, chia, trong đó phép chia thực hiện trên số nguyên.

Input đảm bảo kết quả phép toán có giá trị tuyệt đối không vượt quá 1018.

**Output:**

Với mỗi test, in ra giá trị biểu thức tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 3  7  2 3 1 \* + 9 –  7  - + 8 \* 7 5 9  3  -10 -4 + | -4  34  -14 |

**BÀI 9. DSA\_P056. SỐ NGUYÊN THUỶ**

Cho số nguyên N. Nhiệm vụ của bạn hãy đưa ra N số nguyên thủy đầu tiên theo thứ tự từ nhỏ đến lớn.

Số K được gọi là số nguyên thủy nếu số đó thỏa mãn tất cả các điều kiện:

* Số các chữ số của K là một số chẵn.
* Tất cả các chữ số của K chỉ bao gồm số 4 hoặc 5.
* K là một số đối xứng.

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test được là một số N được viết trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤104.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  10 | 44 55 4444 4554  44 55 4444 4554 5445 5555 444444 445544 454454 455554 |

**BÀI 10. DSA\_P057. SỐ LỘC PHÁT ĐỐI XỨNG CHẴN**

Trong quan điểm người Việt, một số nguyên dương được gọi là “lộc phát” nếu chỉ có hai chữ số là 6 và/hoặc 8. Một số lộc phát X được gọi là đối xứng chẵn nếu:

* Số chữ số của X là chẵn
* X thỏa mãn tính chất đối xứng.

Hãy liệt kê N số lộc phát đối xứng chẵn đầu tiên theo thứ tự từ nhỏ đến lớn.

**Input**

* Dòng đầu tiên đưa vào số lượng test T.
* Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test được là một số N được viết trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100; 1≤N≤104.

**Output**

Đưa ra kết quả mỗi test theo từng dòng. Các số cách nhau đúng một khoảng trống.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4  10 | 66 88 6666 6886  66 88 6666 6886 8668 8888 666666 668866 686686 688886 |

**BÀI 11. DSA\_P059. SỐ 2 ƯU THẾ**

Hệ thống máy tính mới chuyển sang sử dụng hệ đếm tam phân với ba chữ số 0, 1, 2.

Do vốn đã quen với hệ đếm nhị phân nên Nam chỉ quan tâm đến các số tam phân thỏa mãn chữ số 2 chiếm ưu thế, tức là số lượng chữ số 2 chiếm nhiều hơn 50% số chữ số của số đó.

Hãy giúp Nam liệt kê N số tam phân ưu thế 2 đầu tiên.

**Input**  
Dòng đầu ghi số bộ test (không quá 20)

Mỗi bộ test ghi số nguyên dương N (không quá 1000)

**Output**

Với mỗi test, viết trên một dòng N số tam phân ưu thế 2, các số cách nhau một khoảng trống.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5  10 | 2 22 122 202 212  2 22 122 202 212 220 221 222 1222 2022 |

**BÀI 12. DSA\_P058. SỐ BƯỚC DI CHUYỂN ÍT NHẤT**

Cho ma trận A kích thước N\*M.

Hãy tìm **số bước đi ít nhất** để di chuyển từ vị trí A[1][1] đến vị trí A[N][M].

Biết rằng mỗi bước từ vị trí (i, j) ta có thể di chuyển theo một trong ba hướng:

* Hướng xuống dưới với số ô di chuyển là hiệu hai giá trị A[i][j] và A[i+1][j]
* Hướng sang phải với số ô di chuyển là hiệu hai giá trị A[i][j] và A[i][j+1]
* Hướng chéo xuống với số ô di chuyển là hiệu của hai giá trị A[i][j] và A[i+1][j+1]

**Input:**

* Dòng đầu tiên đưa vào số lượng test T.
* Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là hai số N, M; phần thứ hai là các phần tử của ma trận A[][]; các số được viết cách nhau một vài khoảng trống.
* T, N, M, A[i][j] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N, M, A[i][j]≤103.

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.
* Nếu không tìm được đường đi ghi ra -1

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  3 3  2 1 2  1 2 4  1 3 2 | 3 |

**BÀI 13. DSA\_P069. VUA VÀ QUÂN MÃ**

Vua Titan và các hiệp sĩ bàn tròn thường gặp nhau vào đầu năm mới để ăn mừng tình bạn của họ. Để tưởng nhớ sự kiện này, chúng ta xem xét một trò chơi, trong đó có một quân vua và một quân mã được đặt ngẫu nhiên trên 2 ô riêng biệt. Bàn cờ có kích thước N×N, trên bàn cờ có một số ô cấm những ô còn lại là những ô tự do – ô có thể di chuyển vào được. 2 ô đặt quân mã và quân vua đang đứng ở các ô tự do.

|  |  |
| --- | --- |
| chess king moves for Sale OFF 72% | chess board knight moves for Sale OFF 67% |

Tại mỗi bước tất cả các quân đều phải di chuyển theo quy tắc và không được đi vào ô cấm, hãy tìm cách di chuyển để chúng gặp nhau nhanh nhất.

**Input:**

Dòng đầu tiên là số nguyên N (N ≤ 100).

N dòng tiếp theo, mỗi dòng gồm 1 xâu có N ký tự, mô tả bàn cờ. “.” Thể hiện ô trống, “#” thể hiện ô cấm, “T” thể hiện vị trí vua đang đứng và “M” thể hiện vị trí quân mã đang đứng.

**Output:**

In ra đáp án là số bước ít nhất để các quân gặp nhau. Nếu không thể gặp nhau, in ra -1.

**Ví dụ:**

|  |  |
| --- | --- |
| Input | Output |
| 5  M....  .....  .#...  .#..#  ...#T | 2 |

**BÀI 14. DSA\_P060. CHỮ SỐ NGUYÊN TỐ**

Chúng ta đều biết chỉ có 4 chữ số nguyên tố là 2, 3, 5, 7. Hãy liệt kê tất cả các số có ít nhất 4 chữ số nhưng không quá N chữ số và thỏa mãn tất cả các điều kiện sau:

* Chỉ có các chữ số 2, 3, 5, 7
* Có đầy đủ 4 chữ số 2, 3, 5, 7
* Không phải là số chẵn.

**Input**

Chỉ có 1 dòng ghi số N (3 < N < 10)

**Output**

Ghi ra lần lượt các số thỏa mãn theo thứ tự tăng dần, mỗi số trên một dòng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | 2357  2375  2537  2573  2735  2753  3257  3275  3527  3725  5237  5273  5327  5723  7235  7253  7325  7523 |

**BÀI 15. DSA\_P067. SỐ MAY MẮN TIẾP THEO**

Trong bài tập môn CTDL và GT, số may mắn được hiểu là số chỉ có hai chữ số 4 và 7. Với mỗi số nguyên dương N thì số may mắn tiếp theo của N được định nghĩa là số may mắn nhỏ nhất lớn hơn hoặc bằng N.

Cho hai số nguyên dương a và b (với a <= b). Hãy tính tổng các số may mắn tiếp theo của tất cả các số trong đoạn [a,b]

**Input**

Chỉ có một dòng ghi hai số a,b (1 <= a <= b <= 109).

**Output**

Ghi ra giá trị kết quả tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 7 | 33 |
| 7 7 | 7 |

BÀI GIẢI THAM KHẢO

**BÀI 1. DSA\_P051. ĐẢO TỪ**

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

cin.ignore();

while(t--){

string s;

getline(cin, s);

stringstream ss(s);

string token;

vector<string>v;

while(ss>>token) v.push\_back(token);

for(string x: v){

for(int j = x.size()-1;j>=0;j--) cout<<x[j];

cout<<" ";

}

cout<<endl;

}

}

**BÀI 2. DSA\_P061. KIỂM TRA CÁC CẶP DẤU NGOẶC**

#include<bits/stdc++.h>

using namespace std;

bool valid(char a, char b){

if((a=='('&&b==')')||(a=='['&&b==']') ||(a=='{'&&b=='}')) return true;

return false;

}

bool check(string s){

stack<char>st;

for(char i: s){

if(i=='('||i=='['||i=='{'){

st.push(i);

} else {

if(st.empty()) return false;

if(valid(st.top(),i)) st.pop();

else return false;

}

}

return st.empty();

}

int main(){

int t;

cin>>t;

while(t--){

string s;

cin>>s;

if(check(s)) cout<<"true"<<endl;

else cout<<"false"<<endl;

}

}

**BÀI 3. DSA\_P052. ĐÁNH SỐ THỨ TỰ CẶP DẤU NGOẶC**

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

cin.ignore();

while(t--){

string s;

getline(cin, s);

int cnt = 1;

stack<int>st;

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

if(s[i] == '('){

st.push(cnt);

cout<<cnt<<" ";

cnt++;

} else if (s[i] == ')'){

cout<<st.top()<<" ";

st.pop();

}

}

cout<<endl;

}

}

**BÀI 4. DSA\_P091. THAY THẾ DẤU NGOẶC**

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

cin >> t;

cin.ignore();

while (t--){

string s;

cin >> s;

stack<int> st; // stack nay luu vi tri cac dau sai

int a[1005] = {0}; // a[i] = 1: Ngoac vi tri i dung/vi tri do khong la dau ngoac, a[i] = 0: Ngoac vi tri i sai

for (int i = 0; i < s.size(); i++) a[i] = 0;

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

if (s[i] == '(') {

st.push(i);

} else if (s[i] == ')') {

if (!st.empty()){

// Danh dau dau mo 0 dau dong la 1

a[i] = 1;

a[st.top()] = 0;

st.pop();

} else {

a[i] = -1;

}

}

}

while (!st.empty()){

// Trong stack toan vi tri ngoac sai

int p = st.top();

a[p] = -1;

st.pop();

}

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

if (s[i] == '(' || s[i] == ')') cout << a[i];

else cout << s[i];

}

cout << endl;

}

}

**BÀI 5. DSA\_P220. NHẢY SANG PHẢI**

#include<bits/stdc++.h>

using namespace std;

int main(){

int n, q;

cin>>n>>q;

int a[n];

for(int i = 0;i<n;i++) cin>>a[i];

stack<int>st;

int r[n];

for(int i = n - 1;i>=0;i--){

while(!st.empty()&&a[i]>=a[st.top()]) st.pop();

if(st.empty()) r[i] = -1;

else r[i] = st.top();

st.push(i);

}

while(q--){

int index, buoc = 0;

cin>>index;

index--;

while(r[index]!=-1){

buoc++;

index = r[index];

}

cout<<buoc<<endl;

}

}

**BÀI 6. DSA\_P053. ĐOẠN LIÊN TIẾP**

#include<bits/stdc++.h>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

int n;

cin>>n;

int a[n];

for(int i=0;i<n;i++) cin>>a[i];

stack<int>st;

int d[n];

for(int i = 0;i<n;i++){

while(!st.empty()&&a[i]>=a[st.top()]) st.pop();

if(st.empty()) d[i] = i+1;

else d[i] = i - st.top();

st.push(i);

}

for(int i = 0;i<n;i++) cout<<d[i]<<" ";

cout<<endl;

}

}

**BÀI 7. DSA\_P054. HÌNH CHỮ NHẬT ĐƠN SẮC**

#include<bits/stdc++.h>

using namespace std;

long long kq(long long a[], long long n){

long long l[n], r[n];

// Tìm Vị trí phần tử đầu tiên bên trái, bên phải a[i] bé hơn a[i]

//l, r là cận trái, phải của HCN

for(long long i = 0;i<n;i++) cin>>a[i];

stack<long long>st1, st2;

for(long long i = n - 1;i>=0;i--){

while(!st1.empty() && a[i] <= a[st1.top()]) st1.pop();

if(st1.empty()) r[i] = n - 1;

else r[i] = st1.top() - 1;

st1.push(i);

}

for(long long i = 0;i<n;i++){

while(!st2.empty() && a[i] <= a[st2.top()]) st2.pop();

if(st2.empty()) l[i] = 0;

else l[i] = st2.top() + 1;

st2.push(i);

}

long long res = -1;

for(long long i = 0;i<n;i++){

res = max(res, 1LL \* (r[i] - l[i] + 1) \* a[i]);

}

return res;

}

int main(){

long long m, n;

cin>>m>>n;

long long a[n], b[n];

for(int i = 0;i<n;i++) cin>>a[i], b[i] = m - a[i];

long long res = -1;

res = max(res, kq(a, n));

res = max(res, kq(b, n));

cout<<res<<endl;

}

**BÀI 8. DSA\_P207. TÍNH TOÁN GIÁ TRỊ BIỂU THỨC**

#include<bits/stdc++.h>

using namespace std;

long long calc\_hauTo(long long x, long long y, string s){

if(s=="+") return y + x;

else if (s=="-") return y - x;

else if (s=="\*") return y \* x;

else return y/x;

}

long long calc\_tienTo(long long x, long long y, string s){

if(s=="+") return x + y;

else if (s=="-") return x - y;

else if (s=="\*") return x \* y;

else return x/y;

}

int toantu(string s){

if(s=="+"||s=="-"||s=="\*"||s=="/") return 1;

else return 0;

}

int main(){

int t;

cin>>t;

while(t--){

int n;

cin>>n;

vector<string>v;

while(n--){

string s;

cin>>s;

v.push\_back(s);

}

stack<long long>st;

int check = 0;//0: tiền tố, 1: hậu tố

if(!toantu(v[0])) check = 1;

if(!check){

for(int i = v.size()-1;i>=0;i--){

if(!toantu(v[i])){

long long x = stoll(v[i]);

st.push(x);

} else {

long long x1 = st.top(); st.pop();

long long x2 = st.top(); st.pop();

long long x3 = calc\_tienTo(x1, x2, v[i]);

st.push(x3);

}

}

} else {

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

if(!toantu(v[i])){

long long x = stoll(v[i]);

st.push(x);

} else {

long long x1 = st.top(); st.pop();

long long x2 = st.top(); st.pop();

long long x3 = calc\_hauTo(x1, x2, v[i]);

st.push(x3);

}

}

}

cout<<st.top()<<endl;

}

}

**BÀI 9. DSA\_P056. SỐ NGUYÊN THUỶ**

#include <bits/stdc++.h>

using namespace std;

using ll = long long;

vector<string>v;

void gen() {

queue<string>q;

q.push("4");

q.push("5");

while (1) {

//Lấy đỉnh qe

string top = q.front();

q.pop();

//Tạo xâu đảo ngược

string tmp = top;

reverse(tmp.begin(), tmp.end());

//Xâu mới được tạo thành từ xâu đầu và đảo xâu đó

string x = "";

x += top + tmp;

if (x != "45" && x != "54") v.push\_back(x);

if (v.size() > 10000) break;

q.push(top + "4");

q.push(top + "5");

}

}

int main(){

gen();

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

for (int i = 0; i < n; i++) cout << v[i] << " ";

cout << endl;

}

}

**BÀI 10. DSA\_P057. SỐ LỘC PHÁT ĐỐI XỨNG ĐỘ DÀI CHẴN**

#include <bits/stdc++.h>

using namespace std;

using ll = long long;

vector<string>v;

void gen() {

queue<string>q;

q.push("6");

q.push("8");

while (1) {

//Lấy đỉnh qe

string top = q.front();

q.pop();

//Tạo xâu đảo ngược

string tmp = top;

reverse(tmp.begin(), tmp.end());

//Xâu mới được tạo thành từ xâu đầu và đảo xâu đó

string x = "";

x += top + tmp;

if (x != "68" && x != "86") v.push\_back(x);

if (v.size() > 10000) break;

q.push(top + "6");

q.push(top + "8");

}

}

int main(){

gen();

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

for (int i = 0; i < n; i++) cout << v[i] << " ";

cout << endl;

}

}

**BÀI 11. DSA\_P059. SỐ 2 ƯU THẾ**

#include<bits/stdc++.h>

using namespace std;

string s = "012";

vector<string>v;

int check(string s){

map<char, int>mp;

for(auto i: s) mp[i]++;

if(mp['2']>s.size()/2) return true;

else return false;

}

void gen(){

//Buoc 1: Khoi tao

queue<string>q;

q.push("1"); q.push("2");

v.push\_back("2");

//Buoc 2: Lap

while(1){

string x = q.front();

q.pop();

if(v.size()==1005) break;

for(char i: s){

string s1 = x + i;

q.push(s1);

if(check(s1)) v.push\_back(s1);

}

}

}

int main(){

int t;

cin>>t;

gen();

while(t--){

int n;

cin>>n;

for(int i = 0;i<n;i++) cout<<v[i]<<" ";

cout<<endl;

}

}

**BÀI 12. DSA\_P058. SỐ BƯỚC DI CHUYỂN ÍT NHẤT**

#include <bits/stdc++.h>

using namespace std;

int a[1005][1005], d[1005][1005], visited[1005][1005];//d[x][y] là số bước di chuyển ít nhất từ ô (0, 0) đễn (x, y)

int dx[3] = {1, 0, 1};

int dy[3] = {0, 1, 1};

int main() {

int t;

cin >> t;

while (t--) {

int n, m;

cin >> n >> m;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) cin >> a[i][j];

}

//Bước 1: Khởi tạo qe, đưa cấu hình cơ sở

memset(d, 0, sizeof(d));

memset(visited, 0, sizeof(visited));

queue<pair<int, int>> q;

q.push({0, 0});

visited[0][0] = 1;

d[0][0] = 0;

int check = 0;

//Bước 2: Lấy đỉnh hàng đợi, loang dần

while (!q.empty()) {

pair<int, int> p = q.front();

q.pop();

int i = p.first, j = p.second;

if (i == n - 1 && j == m - 1) {

check = 1;

cout << d[i][j] << endl;

break;

}

int i1, j1, step;

for (int k = 0; k < 3; k++) {

if (k == 0 && j + 1 < m) {

step = abs(a[i][j] - a[i][j + 1]);

i1 = i;

j1 = j + step;

} else if (k == 1 && i + 1 < n) {

step = abs(a[i][j] - a[i + 1][j]);

i1 = i + step;

j1 = j;

} else if (k == 2 && i + 1 < n && j + 1 < m) {

step = abs(a[i][j] - a[i + 1][j + 1]);

i1 = i + step;

j1 = j + step;

}

if (i1 >= 0 && i1 < n && j1 >= 0 && j1 < m && visited[i1][j1] == 0) {

q.push({i1, j1});

d[i1][j1] = d[i][j] + 1;

visited[i1][j1] = 1;

}

}

}

if(!check) cout<<-1<<endl;

}

}

**BÀI 13. DSA\_P069. VUA VÀ QUÂN MÃ**

#include<bits/stdc++.h>

using namespace std;

using ll = long long;

int n,xm,ym,xv,yv;

char a[1005][1005];

bool flag[1005][1005];

bool visited[1005][1005];

int dp[1005][1005];

const int mx[8] = {-1,-2,-2,-1,1,2,2,1};

const int my[8] = {-2,-1,1,2,2,1,-1,-2};

int dp1[1005][1005];

const int vx[8] = {1,-1,0,0,1,-1,1,-1};

const int vy[8] = {1,1,1,-1,-1,-1,0,0};

void BFS(int i, int j){

queue <pair<int,int>> q;

q.push({i,j});

visited[i][j] = true;

while(!q.empty()){

pair<int,int> x = q.front();

q.pop();

for(int k = 0; k < 8; k++){

int i1 = x.first + mx[k], j1 = x.second + my[k];

if(i1 >= 1 && i1 <= n && j1 >= 1 && j1 <= n && !visited[i1][j1]){

dp[i1][j1] = dp[x.first][x.second] + 1;

q.push({i1,j1});

visited[i1][j1] = true;

}

}

}

}

void BFS1(int i, int j){

queue <pair<int,int>> q;

q.push({i,j});

flag[i][j] = true;

while(!q.empty()){

pair<int,int> x = q.front();

q.pop();

for(int k = 0; k < 8; k++){

int i1 = x.first + vx[k], j1 = x.second + vy[k];

if(i1 >= 1 && i1 <= n && j1 >= 1 && j1 <= n && !flag[i1][j1]){

dp1[i1][j1] = dp1[x.first][x.second] + 1;

q.push({i1,j1});

flag[i1][j1] = true;

}

}

}

}

int main(){

cin >> n;

for(int i = 1; i <= n; i++){

for(int j = 1; j <= n; j++){

cin >> a[i][j];

if(a[i][j] == '#') {

flag[i][j] = true;

visited[i][j] = true;

}

else if(a[i][j] == 'M'){

xm = i; ym = j;

}

else if(a[i][j] == 'T'){

xv = i; yv = j;

}

}

}

int ans = 1e6;

BFS(xm,ym);

BFS1(xv,yv);

for(int i = 1; i <= n; i++){

for(int j = 1; j <= n; j++){

if(dp[i][j] == dp1[i][j] && dp[i][j] != 0){

ans = min(ans,dp[i][j]);

}

}

cout << endl;

}

cout << ans << endl;

}

**BÀI 14. DSA\_P060. CHỮ SỐ NGUYÊN TỐ**

#include<bits/stdc++.h>

using namespace std;

int n; string s = "2357"; vector<string>v;

int check(string s){

if(s.back()=='2') return 0;

set<char>sc;

for(auto i: s) sc.insert(i);

if(sc.size()==4) return 1;

else return 0;

}

void gen(){

//Buoc 1: Khoi tao

queue<string>q;

for(char i: s){

string tmp = ""; tmp+=i;

q.push(tmp);

}

//Buoc 2: Lap

while(1){

string x = q.front();

q.pop();

if(x.size()==n) break;

for(char i: s){

string s1 = x + i;

q.push(s1);

if(check(s1)&&s1.size()>=4) v.push\_back(s1);

}

}

}

int main(){

cin>>n;

gen();

for(auto i: v) cout<<i<<endl;

}

**BÀI 15. DSA\_P067. SỐ MAY MẮN TIẾP THEO**

#include <bits/stdc++.h>

using namespace std;

int main() {

int a, b;

cin >> a >> b;

queue<long long> q;

q.push(4);

q.push(7);

long long kq = 0;

int i = a;

while (1) {

long long tmp = q.front();

q.pop();

while (i <= tmp && i <= b) {

kq += tmp;

i++;

}

if (i > b) break;

q.push(tmp \* 10 + 4);

q.push(tmp \* 10 + 7);

}

cout << kq;

}