Week 10

=====================================================

Problem 1001:

=====================================================

#include <iostream>

using namespace std;

struct Train

{

int ID;

string destination\_address;

string departure\_time;

};

int main()

{

int N;

cin >> N;

Train\* Travel = new Train[N];

double time\_variable = 0;

for (Train\* it = Travel; it != Travel + N; it++)

{

it->destination\_address = new char[21];

it->departure\_time = new char[21];

cin >> it->ID >> it->destination\_address >> it->departure\_time;

}

for (Train\* it = Travel; it != Travel + N; it++)

for (int j = 0; j < it->destination\_address.length(); j++)

if (it->destination\_address[j] >= 'a' && it->destination\_address[j] <= 'z')

it->destination\_address[j] = it->destination\_address[j] - 32;

string city;

cin >> city;

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

if (city[i] >= 'a' && city[i] <= 'z')

city[i] = city[i] - 32;

int k = 0;

for (Train\* it = Travel; it != Travel + N; it++)

{

if (city == it->destination\_address)

{

cout << it->ID << "\_" << it->destination\_address << "\_" << it->departure\_time;

k++;

}

}

if (k == 0)

cout << "Impossible";

}

=====================================================

Problem 1002:

=====================================================

#include <iostream>

using namespace std;

struct Train

{

int ID;

string destination\_address;

string departure\_time;

};

int main()

{

int N;

cin >> N;

Train\* Travel = new Train[N];

for (Train\* it = Travel; it != Travel + N; it++)

cin >> it->ID >> it->destination\_address >> it->departure\_time;

for (Train\* it = Travel; it != Travel + N; it++)

for (int j = 0; j < it->destination\_address.length(); j++)

if (it->destination\_address[j] >= 'a' && it->destination\_address[j] <= 'z')

it->destination\_address[j] = it->destination\_address[j] - 32;

string city;

cin >> city;

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

if (city[i] >= 'a' && city[i] <= 'z')

city[i] = city[i] - 32;

int k = 0;

string time\_variable = "23:59";

for (Train\* it = Travel; it != Travel + N; it++)

if (city == it->destination\_address)

if (time\_variable > it->departure\_time)

time\_variable = it->departure\_time;

for (Train\* it = Travel; it != Travel + N; it++)

if (city == it->destination\_address && time\_variable == it->departure\_time)

{

cout << it->ID << "\_" << it->destination\_address << "\_" << time\_variable;

k++;

}

if (k == 0)

cout << "Impossible";

}

=====================================================

Problem 1003:

=====================================================

#include <iostream>

using namespace std;

struct Student

{

int ID;

char\* surname;

double grade;

};

int main()

{

int N;

cin >> N;

Student\* Aitu = new Student[N];

double maximumm = 0;

for (Student\* it = Aitu; it != Aitu + N; it++)

{

it->surname = new char[21];

cin >> it->ID >> it->surname >> it->grade;

}

for (Student\* it = Aitu; it != Aitu + N; it++)

if (it->grade > maximumm)

maximumm = it->grade;

for (Student\* it = Aitu; it != Aitu + N; it++)

if (maximumm == it->grade)

cout << it->ID << " " << it->surname << " " << it->grade << endl;

}

=====================================================

Problem 1004:

=====================================================

#include <iostream>

using namespace std;

struct Student

{

int ID;

double grade;

};

int main()

{

int N;

cin >> N;

Student\* Aitu = new Student[N];

double time\_variable = 0;

for (Student\* it = Aitu; it != Aitu + N; it++)

{

cin >> it->ID >> it->grade;

}

for (int i = 0; i < N - 1; i++)

for (int j = 0; j < N - 1; j++)

{

if ((Aitu + j)->grade < Aitu[j + 1].grade)

swap(Aitu[j], \*(Aitu + j + 1));

else if ((Aitu + j)->grade == Aitu[j + 1].grade && (Aitu + j)->ID > Aitu[j + 1].ID)

swap(Aitu[j], \*(Aitu + j + 1));

}

for (Student\* it = Aitu; it != Aitu + N; it++)

cout << it->ID << " " << " " << it->grade << endl;

}

=====================================================

Problem 1005:

=====================================================

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

struct Luggage

{

char description[100];

int count[10];

double weight[10];

};

struct Passenger

{

int luggageSize;

char name[100];

Luggage luggage;

};

int main()

{

int n;

cin >> n;

Passenger\* person = new Passenger[n];

string name;

string fruit;

double max\_weight = 0;

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

{

cin >> person[i].name;

cin >> person[i].luggageSize;

for (int j = 0; j < person[i].luggageSize; j++)

{

cin >> person[i].luggage.description;

cin >> person[i].luggage.count[j];

cin >> person[i].luggage.weight[j];

if (person[i].luggage.weight[j] > max\_weight)

{

name = person[i].name;

fruit = person[i].luggage.description;

max\_weight = person[i].luggage.weight[j];

}

}

}

cout << name << endl << fruit;

return 0;

}

=====================================================

Problem 1006:

=====================================================

#include <iostream>

using namespace std;

struct Train

{

int ID;

string FromAddress;

string ToAdress;

string DepartureTime;

};

bool function(string adress\_1, string adress\_2)

{

bool answer;

int size1 = adress\_1.length();

int size2 = adress\_2.length();

for (int i = 0; i < min(size1, size2); i++)

{

if (adress\_1[i] > adress\_2[i])

{

answer = true;

break;

}

else if (adress\_1[i] < adress\_2[i])

{

answer = false;

break;

}

else

continue;

}

return answer;

}

int main()

{

int N;

cin >> N;

Train\* travel = new Train[N];

for (int i = 0; i < N; i++)

cin >> travel[i].ID >> travel[i].FromAddress >> travel[i].ToAdress >> travel[i].DepartureTime;

for (int i = 0; i < N; i++)

for (int j = 0; j < travel[i].ToAdress.length(); j++)

if ('a' <= travel[i].ToAdress[j] && travel[i].ToAdress[j] <= 'z')

travel[i].ToAdress[j] = travel[i].ToAdress[j] - 32;

for (int i = 0; i < N; i++)

for (int j = 0; j < travel[i].FromAddress.length(); j++)

if ('a' <= travel[i].FromAddress[j] && travel[i].FromAddress[j] <= 'z')

travel[i].FromAddress[j] = travel[i].FromAddress[j] - 32;

for (int i = 0; i < N - 1; i++)

for (int j = 0; j < N - 1; j++)

if (function(travel[j].ToAdress, travel[j + 1].ToAdress))

swap(travel[j], travel[j + 1]);

for (int i = 0; i < N; i++)

cout << travel[i].ID << " " << travel[i].FromAddress << " " << travel[i].ToAdress << " " << travel[i].DepartureTime << endl;

}

=====================================================

Problem 1007:

=====================================================

#include <iostream>

#include <string>

using namespace std;

struct Students

{

string ID;

string Name;

string Surname;

int Subject1;

int Subject2;

int Subject3;

int SelectedSubject;

string SpecialCase;

};

void function(Students\* Winers, int N, int M, int special\_count)

{

int count = 0;

for (int i = 0; i < N; i++)

{

if ((count < M - special\_count && Winers[i].SpecialCase == "NO"))

{

cout << Winers[i].ID << " " << Winers[i].Name << " " << Winers[i].Surname << " " << Winers[i].Subject1 + Winers[i].Subject2 + Winers[i].Subject3 + Winers[i].SelectedSubject << endl;

count++;

}

if (Winers[i].SpecialCase == "YES")

cout << Winers[i].ID << " " << Winers[i].Name << " " << Winers[i].Surname << " " << Winers[i].Subject1 + Winers[i].Subject2 + Winers[i].Subject3 + Winers[i].SelectedSubject << endl;

}

}

int main()

{

int N, M, special\_count = 0;

cin >> N >> M;

Students\* Winers = new Students[N];

for (int i = 0; i < N; i++)

cin >> Winers[i].ID >> Winers[i].Name >> Winers[i].Surname >> Winers[i].Subject1 >> Winers[i].Subject2 >> Winers[i].Subject3 >> Winers[i].SelectedSubject >> Winers[i].SpecialCase;

for (int i = 0; i < N; i++)

if (Winers[i].SpecialCase == "YES")

special\_count++;

for (int i = 0; i < N - 1; i++)

for (int j = 0; j < N - 1; j++)

{

if ((Winers[j].Subject1 + Winers[j].Subject2 + Winers[j].Subject3 + Winers[j].SelectedSubject) < (Winers[j + 1].Subject1 + Winers[j + 1].Subject2 + Winers[j + 1].Subject3 + Winers[j + 1].SelectedSubject))

swap(Winers[j], Winers[j + 1]);

else if ((Winers[j].Subject1 + Winers[j].Subject2 + Winers[j].Subject3 + Winers[j].SelectedSubject) == (Winers[j + 1].Subject1 + Winers[j + 1].Subject2 + Winers[j + 1].Subject3 + Winers[j + 1].SelectedSubject))\

if (Winers[j].SelectedSubject < Winers[j + 1].SelectedSubject)

swap(Winers[j], Winers[j + 1]);

}

function(Winers, N, M, special\_count);

}

=====================================================

Problem 1010:

=====================================================

#include <iostream>

#include <string>

using namespace std;

int main()

{

int count\_read = 0, count\_black = 0, count\_green = 0, count\_white = 0;

char stroka[8][8];

for (int i = 0; i < 8; i++)

for (int j = 0; j < 8; j++)

cin >> stroka[i][j];

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 8; j++)

{

if (stroka[i][j] == '0')

count\_black++;

if (stroka[i][j] == '1')

count\_white++;

if (stroka[i][j] == '2')

count\_read++;

if (stroka[i][j] == '3')

count\_green++;

}

}

if (count\_green == 0 || count\_read == 0 || count\_white == 0 || count\_black == 0)

cout << "BAD INPUT LIST";

else

{

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 8; j++)

{

if (stroka[i][j] == '2')

cout << '2';

else

cout << '-';

}

cout << endl;

}

cout << endl;

cout << count\_black << " " << count\_white << " " << count\_read << " " << count\_green;

}

}

=====================================================

Problem 1011:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

int count = 0;

char stroka[100000];

cin >> stroka;

for (int i = 0; i < strlen(stroka); i++)

{

if (stroka[i] == '(' && count > -1)

count++;

else if (stroka[i] == ')' && count > -1)

count--;

else

break;

}

(count == 0) ? cout << "VALID" : cout << "INVALID";

}

=====================================================

Problem 1012:

=====================================================

#include<iostream>

#include<stack>

#include<string>

using namespace std;

bool Check(char opening, char closing)

{

if (opening == '(' && closing == ')') return true;

else if (opening == '{' && closing == '}') return true;

else if (opening == '[' && closing == ']') return true;

return false;

}

bool function(string exp, int N)

{

stack<char> S;

for (int i = 0; i < N; i++)

{

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

S.push(exp[i]);

else if (exp[i] == ')' || exp[i] == '}' || exp[i] == ']')

{

if (S.empty() || !Check(S.top(), exp[i]))

return false;

else

S.pop();

}

}

return S.empty() ? true : false;

}

int main()

{

int N;

cin >> N;

string expression;

cin >> expression;

if (function(expression, N))

cout << "Yes";

else

cout << "No";

}