**Member functions**

[**(constructor)**](https://www.cplusplus.com/reference/bitset/bitset/bitset/)

Construct bitset (public member function )

[**applicable operators**](https://www.cplusplus.com/reference/bitset/bitset/operators/)

Bitset operators (function )

**Bit access**

[**operator[]**](https://www.cplusplus.com/reference/bitset/bitset/operator%5b%5d/)

Access bit (public member function )

[**count**](https://www.cplusplus.com/reference/bitset/bitset/count/)

Count bits set (public member function )

[**size**](https://www.cplusplus.com/reference/bitset/bitset/size/)

Return size (public member function )

[**test**](https://www.cplusplus.com/reference/bitset/bitset/test/)

Return bit value (public member function )

[**any**](https://www.cplusplus.com/reference/bitset/bitset/any/)

Test if any bit is set (public member function )

[**none**](https://www.cplusplus.com/reference/bitset/bitset/none/)

Test if no bit is set (public member function )

[**all**](https://www.cplusplus.com/reference/bitset/bitset/all/)

Test if all bits are set (public member function )

**Bit operations**

[**set**](https://www.cplusplus.com/reference/bitset/bitset/set/)

Set bits (public member function )

[**reset**](https://www.cplusplus.com/reference/bitset/bitset/reset/)

Reset bits (public member function )

[**flip**](https://www.cplusplus.com/reference/bitset/bitset/flip/)

Flip bits (public member function )

**Bitset operations**

[**to\_string**](https://www.cplusplus.com/reference/bitset/bitset/to_string/)

Convert to string (public member function )

[**to\_ulong**](https://www.cplusplus.com/reference/bitset/bitset/to_ulong/)

Convert to unsigned long integer (public member function )

[**to\_ullong**](https://www.cplusplus.com/reference/bitset/bitset/to_ullong/)

Convert to unsigned long long (public member function )

**Non-member function overloads**

[**applicable operators**](https://www.cplusplus.com/reference/bitset/bitset/operators/)

Bitset operators (function )

**Non-member class specializations**

[**hash<bitset>**](https://www.cplusplus.com/reference/bitset/bitset/hash/)

Hash for bitset (class template specialization )

### XẾP HẬU

Cho một bàn cờ 8 x 8, mỗi ô có một giá trị A[i][j] nhất định (0 <= A[i][j] <= 100), tương ứng

với điểm số đạt được nếu như bạn đặt một quân cờ vào đó.

Nhiệm vụ của bạn là đặt 8 quân hậu lên bàn cờ, sao cho không có 2 quân nào tự ăn nhau,

và số điểm đạt được là lớn nhất.

Input: Dòng đầu tiên là số lượng bộ test T (T <= 20).

Mỗi test gồm 8 dòng, mỗi dòng 8 số nguyên mô tả bàn cờ.

Output: Với mỗi test, in ra đáp án trên một dòng.

Ví dụ: QUEEN.INP

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

17 18 19 20 21 22 23 24

25 26 27 28 29 30 31 32

33 34 35 36 37 38 39 40

41 42 43 44 45 46 47 48

49 50 51 52 53 54 55 56

57 58 59 60 61 62 63 64

Output: 260

**Algorithm**

Dữ liệu nhỏ, n = 8 nên vét cạn: xét mọi hoán vị của 1..8

**Complexity**: O(n!), n = số hậu

**Program**

/\*

Name: QUEENS.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

Xep Hau toi uu

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 8;

int a[MN+1][MN+1]; // award

int p[MN+1];

int score = 0;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void ReadData(const char\* fn) {

ifstream f(fn);

if (f.fail()) {

cerr << "\n No flie " << fn;

exit(0);

}

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

cout << endl;

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

f >> a[i][j];

cout << " " << a[i][j];

}

}

f.close();

}

bool Next() {

int i, j;

for(i = MN-1; i > 0; --i)

if(p[i] < p[i+1]) break;

if(i < 1) return false;

for(j = MN; j > i; --j)

if(p[j] > p[i]) break;

int x = p[i]; p[i] = p[j]; p[j] = x;

++i; j = MN;

while(i < j) {

x = p[i]; p[i] = p[j]; p[j] = x;

++i; --j;

}

return true;

}

int GetScore() {

int sc = 0; // score

for(int i = 1; i <= MN; ++i) // Hau i

// Hau i dung tai dong p[i]

// se co thuong a[i][p[i]]

sc += a[i][p[i]];

return sc;

}

bool Sat() {

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

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

if (abs(p[i]-p[j]) == i-j) return false;

}

}

return true;

}

void Run() {

// init permutation p

for(int i = 0; i <= MN; ++i) p[i] = i;

int d = 0, sc;

score = GetScore();

while (Next()) {

if (Sat()) score = max(score, GetScore());

}

cout << "\n Result: " << score;

}

main() {

ReadData("QUEENS.INP");

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### Club

Một lớp có N bạn học sinh mang bí danh lần lượt là A, B, C,…Các bạn học sinh ghi danh vào bốn nhóm chuyên đề Toán, Tin, Văn và Lý.

Hãy hiển thị các danh sách sau đây:

a) Những bạn tham gia cả bốn nhóm

b) Những bạn chỉ tham gia duy nhất nhóm Tin

Input: text file CLUB.INP

Dòng đầu tiên N số học tinh với tên lần lượt là A, B, ...

Tiếp theo là bốn dòng dạng string gồm các ký tự viết liền nhau:

Các học sinh tham gia nhóm Toán

Các học sinh tham gia nhóm Tin

Các học sinh tham gia nhóm Văn

Các học sinh tham gia nhóm Lý

Output: hiển thị trên màn hình

Ví dụ

|  |  |
| --- | --- |
| CLUB.INP | OUTPUT |
| 25  VDAEBCM  TAMBCDVEP  CABEFV  FABVE | a) HS tham gia cả 4 club: ABEV  b) HS tham gia duy nhất club TIN: PT |

#### Thuật toán

Pha 1. Đọc dữ liệu và khởi tạo 4 tập ứng với mỗi club TOÁN, TIN, VĂN, LÝ

Pha 2:

Câu a: Tìm giao của 4 tập

Câu b: Tìm hiệu của tập TIN với ba tập còn lại

#### Chương trình

// Club

#include<bits/stdc++.h>

using namespace std;

const char \* FN = "CLUB.INP";

const int MN = 26;

int n = MN;

// e trong khoang 0..n

#define Inrange(e) (e) >= 0 && (e) < n

// e trong khoang 'A'..'A'+n

#define InABC(c) (c) >= 'A' && (c) <= Chr(n)

// chuyen doi i -> char

#define Chr(i) char('A'+(i))

// chuyen doi char -> i

#define Id(e) int((e)-'A')

// Cau truc du lieu cua Set of CHAR

typedef bitset<MN> Set;

inline void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

inline void Print(const Set &x, const char \* msg = "") { // Hien thi

cout << msg;

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

if (x[i]) cout << Chr(i);

}

inline void Ins(Set &x, char e) { // Nap ki tu e vao Set x

if (InABC(e)) x[Id(e)] = 1;

}

inline void Ins(Set &x, string s) { // Nap string s vao Set x

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

Ins(x,s[i]);

}

inline bool Elem(char e, Set &x) { return x[Id(e)]; }

inline int Num(Set &x) { return x.count(); } // So phan tu hien co

inline int Capacity(Set &x) { return x.size(); } // Kich thuoc toi da

// Hop hai Set a|b

inline Set Union(Set &a, Set &b) { return a | b; }

// Giao hai Set a&b

inline Set Intersection(Set &a, Set &b) { return a & b; }

// Hieu hai Set a-b

inline Set Difference(Set &a, Set &b) { return a & (~b); }

// a <= b

inline bool Sub(Set &a, Set &b) { return (a & b) == a; }

const string CLUB[4] = {"TOAN", "TIN", "VAN", "LY"};

const int TOAN = 0, TIN = 1, VAN = 2, LY = 3;

Set t[4]; // 4 nhom TOAN = 0, TIN = 1, VAN = 2, LY = 3;

void ReadData() {

ifstream f(FN);

if(f.fail()) {

cerr << "\n Khong mo duoc input file " << FN;

exit(1);

}

f >> n;

cout << "\n So HS " << n;

string s;

for(int i = 0; i < 4; ++i) {

f >> s;

// cout << "\n " << CLUB[i] << ": " << s;

Ins(t[i], s);

cout << "\n Club " << CLUB[i] << ": "; Print(t[i], " ");

}

f.close();

}

// Cau a: danh sach cac ban tham gia ca 4 club

// Giao cua 4 Set

Set Caua() {

Set r = t[0];

for(int i = 1; i < 4; ++i)

r = Intersection(r, t[i]);

return r;

}

// Cau b: danh sach cac ban tham gia duy nhat club TIN

// Hieu cua Set TIN vói cac Set con lai

Set Caub() {

Set r = t[TIN];

for(int i = 0; i < 4; ++i)

if (i != TIN)

r = Difference(r, t[i]);

return r;

}

void Run() {

ReadData();

Print(Caua(), "\n a) HS tham gia ca 4 club: ");

Print(Caub(), "\n b) HS tham gia duy nhat club TIN: ");

}

main() {

Run();

cout << "\n T h e E n d";

return 0;

}

#### Output

So HS 25

Club TOAN: ABCDEMV

Club TIN: ABCDEMPTV

Club VAN: ABCEFV

Club LY: ABEFV

a) HS tham gia ca 4 club: ABEV

b) HS tham gia duy nhat club TIN: PT

T h e E n d