

```

// Antroji programavimo užduotis
#include <fstream>
#include <iostream>
using namespace std;
struct kambarys{ // Struktūra pradiniam duomenim apie pabėgimo kambarį išsaugoti
    string diena;
    int valandu_sk;
    int Valandos[24];
    int Tinka[24] = {0}; // Kintamasis naudojamas sekti kiek kiekvienos dienos
    kiekvieną valandą draugų galėtų dalyvauti
};

struct draugai{ // Struktūra pradiniam duomenim apie draugus išsaugoti
    string vardas;
    int variantu_sk;
    string Diena[168];
    int Valanda[168];
};

struct atrinkti_laikai{ // Struktūra išsauganti galutinį atrinktų valandų ir
    galinčių dalyvauti draugų sąrašą
    string diena;
    int valanda;
    int kiekis;
    string Vardai[20];
};

void nuskaitymas(int &d, int &dr, kambarys K[], draugai D[]); // Duomenų iš failo
    nuskaitymo funkcija
void skaiciavimas(int d, int dr, kambarys K[], draugai D[]); // Funkcija,
    apskaičiuojanti kiek mokinių tinka kiekvienos dienos kiekviena valanda
void atrinkimas(int d, kambarys K[], int &a, atrinkti_laikai A[], int dr, draugai
    D[]); // Funkcija, atrenkanti tas dienas, kai draugų buvo ne mažiau kaip keturi,
    ir atrenkanti jų vardus
void rikiavimas(int a, atrinkti_laikai A[]); // Funkcija rikiuojanti tinkamų
    laikų bei mokinių sąrašus
void isvedimas(int a, atrinkti_laikai A[]); // Duomenų išvedimo į failą funkcija

int main()
{
    int d, dr, a = 0;
    kambarys K[7];
    draugai D[20];
    atrinkti_laikai A[168];
    nuskaitymas(d, dr, K, D);
    skaiciavimas(d, dr, K, D);
}

```

```

    atrinkimas(d, K, a, A, dr, D);
    rikiavimas(a, A);
    isvedimas(a, A);

    return 0;
}

void nuskaitymas(int &d, int &dr, kambarys K[], draugai D[]){
    ifstream fd("U2.txt");
    char T1[5], T2[14]; // Laikini kintamieji simbolių nuskaitymui
    fd >> d >> dr;
    for(int i = 0; i < d; i++){
        fd.ignore();
        fd.get(T1, 4);
        K[i].diena = T1;
        fd >> K[i].valandu_sk;
        for (int j = 0; j < K[i].valandu_sk; j++){
            fd >> K[i].Valandos[j];
        }
    }

    for(int i = 0; i < dr; i++){
        fd.ignore();
        fd.get(T2, 13);
        D[i].vardas = T2;
        fd >> D[i].variantu_sk;
        for (int j = 0; j < D[i].variantu_sk; j++){
            fd.ignore();
            fd.get(T1, 4);
            D[i].Diena[j] = T1;
            fd >> D[i].Valanda[j];
        }
    }

    fd.close();
}

void skaiciavimas(int d, int dr, kambarys K[], draugai D[]){
    for (int i = 0; i < d; i++){
        for(int j = 0; j < K[i].valandu_sk; j++){
            for(int l = 0; l < dr; l++){
                for (int k = 0; k < D[l].variantu_sk; k++){
                    // Jei kažkuri diena ir valanda tinka konkrečiam žmogui, tai
                    // padidiname tokių žmonių skaičių vienetu prie atitinkamos dienos valandos
                    if(K[i].diena == D[l].Diena[k] && K[i].Valandos[j] ==
D[l].Valanda[k]){

```

```

        K[i].Tinka[j]++;
    }
}
}
}
}

void atrinkimas(int d, kambarys K[], int &a, atrinkti_laikai A[], int dr, draugai D[]){
    // Atrenkame konkrečių valandų sąrašą
    for (int i = 0; i < d; i++){
        for (int j = 0; j < K[i].valandu_sk; j++){
            if (K[i].Tinka[j] >= 4){
                A[a].diena = K[i].diena;
                A[a].valanda = K[i].Valandos[j];
                A[a].kiekis = K[i].Tinka[j];
                a++;
            }
        }
    }
    int temp; // Laikinas kintamasis, skirtas vardams į struktūrą įrašyti
    for(int i = 0; i < a; i++){
        temp = 0;
        for(int j = 0; j < dr; j++){
            for(int l = 0; l < D[j].variantu_sk; l++){
                // Jei randame, kad konkrečiam žmogui tinka atrinkta diena ir
                valanda, įsirašome jo vardą į sąrašą
                if(A[i].diena == D[j].Diena[l] && A[i].valanda ==
                D[j].Valanda[l]){
                    A[i].Vardai[temp] = D[j].vardas;
                    temp++;
                }
            }
        }
    }
}

void rikiavimas(int a, atrinkti_laikai A[]){
    // Pirma surikiuojame atrinktų valandų sąrašą
    for(int i = 0; i < a-1; i++){
        for(int j = i+1; j < a; j++){
            if(A[i].kiekis < A[j].kiekis){
                swap(A[i], A[j]);
            }
        }
    }
}

```

```

// Tuomet surikiuojame kiekvienos valandos vardų sąrašą
for(int i = 0; i < a; i++){
    for (int j = 0; j < A[i].kiekis - 1; j++){
        for(int l = j+1; l < A[i].kiekis; l++){
            if(A[i].Vardai[j] > A[i].Vardai[l]){
                swap(A[i].Vardai[j], A[i].Vardai[l]);
            }
        }
    }
}

void isvedimas(int a, atrinkti_laikai A[]){
    ofstream fr("U2rez.txt");
    for(int i = 0; i < a; i++){
        fr << A[i].diena << " " << A[i].valanda << " " << A[i].kiekis << endl;
        for(int j = 0; j < A[i].kiekis; j++){
            fr << A[i].Vardai[j] << endl;
        }
    }
    fr.close();
}

```