1. **Užduotis.** **Kambarių sienų dažymas**

Parašykite programą, kuri apskaičiuoja, kiek litrų dažų prireiks padengti kiekvieno kambario sienas 1 kartą ir aptinka didžiausią kiekį dažų, reikalingą kambario sienų padengimui tarp visų dažomų kambarių. (rezultatas pateikiamas dviejų skaitmenų po kablelio tikslumu)

Planuojama dažyti **n** kambarių sienas neskaitant lubų. Nuperkami dažai su **k** m²/l išeiga. Žinoma, kiek skirtingų sienų **s** yra kiekviename kambaryje, taip pat ir jų ilgiai bei aukščiai metrais, kambario durų skaičius **d** bei langų skaičius **r**. Durų plotas visuose kambariuose yra vienodas ir yra lygus 2 m², langų plotas taip pat ir yra lygus 1.5 m². Durys ir langai nedažomi.

Pirmoje pradinių duomenų failo dazai\_data.txt eilutėje įrašytas sveikasis skaičius kambarių **n** ir dažų išeiga **k**. Toliau **n** kartų surašyti kambarių parametrai: eilutėse yra skirtingų sienų skaičius **s**, kambario durų skaičius **d** bei langų skaičius **r**. Po kiekvienos kambario parametrų eilutės yra **s** kartų surašyti skirtingų sienų ilgiai ir aukščiai. Jei kambarys kvadrato formos, tada skirtingų sienų skaičius **s** = 1, jei stačiakampio, tai skirtingų sienų skaičius **s** = 2. Jei kambarys kito daugiakampio formos, tai skirtingų sienų skaičius **s** > 2.

Rezultatų faile dazai\_res.txt turi būti išvesti kiekvieno kambario sienoms padengti reikalingi dažų kiekiai litrais, paskutinėje eilutėje išvedamas didžiausias iš visų kambarių sienoms padengti reikalingas dažų kiekis.

|  |  |
| --- | --- |
| dazai\_data.txt | dazai\_res.txt |
| 3 6  2 1 1  4 3  5 3  5 2 3  2.5 3  1.5 3  2.7 3  4 3  3.5 3  1 1 2  4 3 | 7.42  8.42  6.52  Didžiausias kiekis: 8.42 |

1. **Reikalavimai**

* Parašykite funkciją **void naujasPlotas(double ilgis, double aukstis, int sienuSkaicius, double & plotas)**, kuri per parametrus grąžina apskaičiuotą naują plotą, pridedant einamos sienos apskačiuotą plotą, atsižvelgiant į sienų skaičių.
* Parašykite funkciją **void arDidziausias(double dazuKiekis, double & didziausiasKiekis)**, kuri per parametrus grąžina naują didžiausio reikiamo dažų kiekio reikšmę, jei paduodamas dažų kiekis didesnis už paduotą didžiausią dažų kiekį.
* Nenaudokite masyvų.
* Skaitymui galima naudoti ciklą cikle, n kartų einant per kiekvieną kambarį ir s kartų per kiekvieno kambario sienas.

1. **Programos tekstas C++ kalba**

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

using namespace std;

void NaujasPlotas(double ilgis, double aukstis, int sienuSkaicius, double &plotas);

void ArDidziausias(double dazuKiekis, double &didziausiasKiekis);

int main(){

int n, s, lang, dur;

double dazkof, ilg, aukst, S = 0, kiek = 0, did = 0;

ifstream df ("dazai\_data.txt");

ofstream rf ("dazai\_res.txt");

df >> n >> dazkof;

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

df >> s >> dur >> lang;

for(int i1 = 0; i1 < s; i1++){

df >> ilg >> aukst;

NaujasPlotas(ilg, aukst, s, S);

}

S = S - dur \* 2 - lang \* 1.5;

kiek = S / dazkof;

rf << fixed << setprecision(2) << kiek << endl;

ArDidziausias(kiek, did);

S = 0;

}

rf << "Didziausias kiekis: " << did;

rf.close();

df.close();

return 0;

}

void NaujasPlotas(double ilgis, double aukstis, int sienuSkaicius, double &plotas)

{

if (sienuSkaicius == 1){

plotas = plotas + 4 \* ilgis \* aukstis;

}

else if (sienuSkaicius == 2){

plotas = plotas + 2 \* ilgis \* aukstis;

}

else{

plotas = plotas + ilgis \* aukstis;

}

}

void ArDidziausias(double dazuKiekis, double &didziausiasKiekis)

{

if (dazuKiekis > didziausiasKiekis)didziausiasKiekis = dazuKiekis;

}