|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
| // Krepsininkai  #include <fstream>  #include <iomanip>  #include <iostream>  using namespace std;  const char CDfv[] = "Duomenys14.txt";  const char CRfv[] = "Rezultatai14.txt";  const int CMax = 30;  //-----------------------------------------------------------------------  struct Mokinys {  string pav; // mokinio vardas  int kiek; // tasku skaicius  };  //---------------------------------------------------------------------------------  void Skaityti(Mokinys A[], int & n);  void Spausdinti(Mokinys A[], int n, string eil);  void Atrinkti(Mokinys A[], int n, Mokinys B[], int & m);  int Suma(Mokinys A[], int n);  //---------------------------------------------------------------------------------  int main()  {  Mokinys A[CMax]; int n; // klases mokiniu duomenys  Mokinys B[CMax]; int m; // atrinktu mokiniu duomenys  Skaityti(A, n);  Atrinkti(A, n, B, m);  string eil = "Mokiniu, surinkusiu ne maziau kaip 15 tasku, sarasas";  Spausdinti(B, m, eil);  ofstream fr (CRfv, ios::app); // srautas atveriamas papildyti  fr << "Klases mokinys vidutiniskai surinko tasku: " << Suma(A,n) / n << endl;  fr << "Atrinkti mokiniai vidutiniskai surinko tasku: " << Suma(B, m) / m << endl;  fr.close();    return 0;  }  //-----------------------------------------------------------------------  // Pradiniu duomenu skaitymas is failo: A – duomenu masyvas, n – duomenu skaicius  void Skaityti(Mokinys A[], int & n)  {  ifstream fd (CDfv);  fd >> n; // perskaitomas klases mokiniu skaicius  for (int i = 0; i < n; i++)  fd >> A[i].pav >> A[i].kiek;  fd.close();  }  //-----------------------------------------------------------------------  // Mokiniu sarasas rasomas i faila, nurodyta konstanta CRfv  // A – duomenu masyvas, n – duomenu skaicius  void Spausdinti(Mokinys A[], int n, string eil)  {  ofstream fr (CRfv);  fr << eil << endl;  fr << "----------------------------" << endl;  fr << " Vardas Taskai " << endl;  fr << "----------------------------" << endl;  for (int i = 0; i < n; i++)  fr << " " << setw(15) << left << A[i].pav << " " << setw(10) << A[i].kiek << endl;  fr << "----------------------------" << endl;  fr.close();  }  //-----------------------------------------------------------------------  // Is masyvo A(n) atrenkami i masyva B(m) mokiniai, varzybose surinkæ ne maziau kaip 15 tasku  void Atrinkti(Mokinys A[], int n, Mokinys B[], int & m)  {  m = 0;  for (int i = 0; i < n; i++)  if (A[i].kiek >= 15) {  B[m] = A[i]; // kopijuojami i-ojo mokinio duomenys is masyvo A i masyvo B pabaiga  m++; // masyvo B irasu skaicius padidejo vienetu  }  }  //-----------------------------------------------------------------------  // Apskaiciuoja ir grazina masyve A(n) esanciu mokiniu surinktu tasku suma  int Suma(Mokinys A[], int n)  {  int s = 0;  for (int i = 0; i < n; i++)  s += A[i].kiek;  return s;  }  //----------------------------------------------------------------------- | // Krepsininkai  #include <fstream>  #include <iomanip>  #include <iostream>  using namespace std;  const char CDfv[] = "Duomenys14.txt";  const char CRfv[] = "Rezultatai14.txt";  const int CMax = 30;  //-----------------------------------------------------------------------  struct Mokinys {  string pav; // mokinio vardas  int kiek; // tasku skaicius  };  //---------------------------------------------------------------------------------  void Skaityti(Mokinys A[], int & n);  void Spausdinti(Mokinys A[], int n, string eil);  void Atrinkti(Mokinys A[], int n, Mokinys B[], int & m);  int Suma(Mokinys A[], int n);  int Max(...);  //---------------------------------------------------------------------------------  int main()  {  Mokinys A[CMax]; int n; // klases mokiniu duomenys  Mokinys B[CMax]; int m; // atrinktu mokiniu duomenys  Skaityti(A, n);  // patikrinti ar duomenų failas nėra tuščias  // patikrinti ar surinko 15 taškų  ...{    ofstream fr(CRfv);    ...      fr.close();  }  return 0;  }  //-----------------------------------------------------------------------  // Pradiniu duomenu skaitymas is failo: A – duomenu masyvas, n – duomenu skaicius  void Skaityti(Mokinys A[], int & n)  {  ifstream fd (CDfv);  fd >> n; // perskaitomas klases mokiniu skaicius  for (int i = 0; i < n; i++)  fd >> A[i].pav >> A[i].kiek;  fd.close();  }  //-----------------------------------------------------------------------  // Mokiniu sarasas rasomas i faila, nurodyta konstanta CRfv  // A – duomenu masyvas, n – duomenu skaicius  void Spausdinti(Mokinys A[], int n, string eil)  {  ofstream fr (CRfv);  fr << eil << endl;  fr << "----------------------------" << endl;  fr << " Vardas Taskai " << endl;  fr << "----------------------------" << endl;  for (int i = 0; i < n; i++)  fr << " " << setw(15) << left << A[i].pav << " " << setw(10) << A[i].kiek << endl;  fr << "----------------------------" << endl;  fr.close();  }  //-----------------------------------------------------------------------  // Is masyvo A(n) atrenkami i masyva B(m) mokiniai, varzybose surinkæ ne maziau kaip 15 tasku  void Atrinkti(Mokinys A[], int n, Mokinys B[], int & m)  {  m = 0;  for (int i = 0; i < n; i++)  if (A[i].kiek >= 15) {  B[m] = A[i]; // kopijuojami i-ojo mokinio duomenys is masyvo A i masyvo B pabaiga  m++; // masyvo B irasu skaicius padidejo vienetu  }  }  //-----------------------------------------------------------------------  // Apskaiciuoja ir grazina masyve A(n) esanciu mokiniu surinktu tasku suma  int Suma(Mokinys A[], int n)  {  int s = 0;  for (int i = 0; i < n; i++)  s += A[i].kiek;  return s;  }  //-----------------------------------------------------------------------  // Apskaiciuoja ir grazina masyve A(n) esanciu mokiniu surinktu tasku suma  int Max(...)  {  ...  }  //----------------------------------------------------------------------- |