|  |  |  |
| --- | --- | --- |
|  | 5  Petras Pirmas 2 25  Jurgis Antras 3 1  Algis Trečias 0 59  Robertas Ketvirtas 3 13  Kasparas Penktas 2 45  6  Petras Didysis 1 25  Jurgis Greitasis 3 11  Algis Skrajūnas 0 56  Robertas Pirmūnas 3 13  Kasparas Eiklusis 2 45  Rimas Batuotas 4 15 |  |
|  | Duomenų failai papildymo  2  Petras Pirmas 2 25  Jurgis Antras 3 10  2  Petras Didysis 1 25  Jurgis Greitasis 3 11  Rezultatas papildymo  Bėgikų finalinio bėgimo sąrašas  ----------------------------------------------------  Vardas ir Parodytas Bėgimo  pavardė laikas takelis  ----------------------------------------------------  Petras Didysis 1 : 25 4  Petras Pirmas 2 : 25 3  Jurgis Antras 3 : 10 2  Jurgis Greitasis 3 : 11 1  --------------------------------------------------- |  |
|  |  |  |
| // Bėgikai  #include <fstream>  #include <iomanip>  #include<iostream>  using namespace std;  const char CDfva[] = "Duomenys1.txt";  const char CDfvb[] = "Duomenys2.txt";  const char CRfv[] = "Rezultatai.txt";  const int CMax = 30;  const int CPav = 20;  //-----------------------------------------------------------------------  struct Sportininkas {  string pav; // vardas ir pavardė  int laikas; // laikas sekundėmis  };  //-----------------------------------------------------------------------  void Skaityti(const char fv[], Sportininkas A[], int & n);  void Spausdinti(Sportininkas A[], int n);  void Rikiuoti(Sportininkas A[], int n);  void Atrinkti(Sportininkas A[], int n, Sportininkas B[], int & m);  //-----------------------------------------------------------------------  int main()  {  Sportininkas A[CMax]; int n; // pradinių duomenų masyvas  Sportininkas B[CMax]; int m = 0; // sąrašas iš pradžių turi būti tuščias (m = 0)  Skaityti(CDfva, A, n);  Rikiuoti(A, n);  Atrinkti(A, n, B, m);  Skaityti(CDfvb, A, n);  Rikiuoti(A, n);  Atrinkti(A, n, B, m);  Rikiuoti (B, m);  Spausdinti(B, m);  return 0;  }  //-----------------------------------------------------------------------  // Skaitomi pradiniai duomenys iš failo į masyvą A(n)  void Skaityti(const char fv[], Sportininkas A[], int & n)  {  ifstream fd (fv);  int min, sek;  char eil[CPav+1];  fd >> n; // bėgikų skaičius  fd.ignore(80, '\n'); // pereinama į kitą eilutę  for (int i = 0; i < n; i++) {  fd.get(eil, CPav); // bėgiko vardas ir pavardė  A[i].pav = eil;  fd >> min >> sek; // bėgiko laikas minutėmis ir sekundėmis  fd.ignore(80, '\n'); // pereinama į kitą eilutę  A[i].laikas = min \* 60 + sek; // apskaičiuojamas laikas sekundėmis  }  fd.close();  }  //-----------------------------------------------------------------------  // Masyvo A(n) duomenys rašomi į failą, nurodytą konstanta CRfv  void Spausdinti(Sportininkas A[], int n)  {  ofstream fr (CRfv);  fr << "Bėgikų finalinio bėgimo sąrašas‌" << endl;  fr << "‌--------------------------------‌" << endl;  fr << "Vardas ir Parodytas Bėgimo" << endl;  fr << "pavardė laikas takelis‌" << endl;  fr << "‌--------------------------------‌" << endl;  for (int i = 0; i < n; i++)  fr << setw(20) << left << A[i].pav << " " << A[i].laikas / 60 << " : " << A[i].laikas % 60 << " " << n - i << endl;  fr << "‌--------------------------------‌" << endl;  fr.close();  }  //-----------------------------------------------------------------------  // Rikiuojami masyvo A(n) duomenys pagal laiką didėjančiai  void Rikiuoti(Sportininkas A[], int n)  {  for (int i = 0; i < n; i++)  for (int j = i + 1; j < n; j++)  if (A[j].laikas < A[i].laikas) {  Sportininkas sp = A[i];  A[i] = A[j];  A[j] = sp;  }  }  //-------------------------------------------------------------------  // Iš masyvo A(n) pusė duomenų kopijuojama į masyvą B(m)  void Atrinkti(Sportininkas A[], int n, Sportininkas B[], int & m)  {  for (int i = 0; i < n / 2; i++ ) {  B[m] = A[i];  m++;  }  } | // Bėgikai  #include <fstream>  #include <iomanip>  #include<iostream>  using namespace std;  const char CDfva[] = "Duomenys3.txt";  const char CDfvb[] = "Duomenys4.txt";  const char CRfv[] = "Rezultatai2.txt";  const int CMax = 30;  const int CPav = 20;  //-----------------------------------------------------------------------  struct Sportininkas {  string pav; // vardas ir pavardė  int laikas; // laikas sekundėmis  };  //-----------------------------------------------------------------------  void Skaityti(const char fv[], Sportininkas A[], int & n);  void Spausdinti(Sportininkas A[], int n);  void Rikiuoti(Sportininkas A[], int n);  void Atrinkti(Sportininkas A[], int n, Sportininkas B[], int & m);  //-----------------------------------------------------------------------  int main()  {  Sportininkas A[CMax]; int n; // pradinių duomenų masyvas  Sportininkas B[CMax]; int m = 0; // sąrašas iš pradžių turi būti tuščias (m = 0)  Skaityti(CDfva, A, n);  Rikiuoti(A, n);  Atrinkti(A, n, B, m);  Skaityti(CDfvb, A, n);  Rikiuoti(A, n);  Atrinkti(A, n, B, m);  Rikiuoti (B, m);  Spausdinti(B, m);  return 0;  }  //-----------------------------------------------------------------------  // Skaitomi pradiniai duomenys iš failo į masyvą A(n)  void Skaityti(const char fv[], Sportininkas A[], int & n)  {  ifstream fd (fv);  int min, sek;  char eil[CPav+1];  fd >> n; // bėgikų skaičius  fd.ignore(80, '\n'); // pereinama į kitą eilutę  for (int i = 0; i < n; i++){  fd.get(eil, CPav); // bėgiko vardas ir pavardė  A[i].pav = eil;  fd >> min >> sek; // bėgiko laikas minutėmis ir sekundėmis  fd.ignore(80, '\n'); // pereinama į kitą eilutę  A[i].laikas = min \* 60 + sek; // apskaičiuojamas laikas sekundėmis  }  fd.close();  }  //-----------------------------------------------------------------------  // Masyvo A(n) duomenys rašomi į failą, nurodytą konstanta CRfv  void Spausdinti(Sportininkas A[], int n)  {  ofstream fr (CRfv);  fr << "Bėgikų finalinio bėgimo sąrašas" << endl;  fr << "--------------------------------" << endl;  fr << "Vardas ir Parodytas Bėgimo" << endl;  fr << "pavardė laikas takelis" << endl;  fr << "--------------------------------" << endl;  for (int i = 0; i < n; i++)  fr << setw(20) << left << A[i].pav << " " << A[i].laikas / 60 << " : " << A[i].laikas % 60 << " " << n - i << endl;  fr << "--------------------------------" << endl;  fr.close();  }  //-----------------------------------------------------------------------  // Rikiuojami masyvo A(n) duomenys pagal laiką didėjančiai  void Rikiuoti(Sportininkas A[], int n)  {  for (int i = 0; i < n; i++)  for (int j = i + 1; j < n; j++)  if (A[j].laikas < A[i].laikas){  Sportininkas sp = A[i];  A[i] = A[j];  A[j] = sp;  }  }  //-------------------------------------------------------------------  // Iš masyvo A(n) pusė duomenų kopijuojama į masyvą B(m)  void Atrinkti(Sportininkas A[], int n, Sportininkas B[], int & m)  {  ...  }  } | // Kolekcija (125-126 psl. užd. Nr.1)  #include <fstream>  #include <iomanip>  #include<iostream>  using namespace std;  const char CDfv[] = "Duomenys16\_u1.txt";  const char CRfv[] = "Rezultatai16\_u1.txt";  const int CMax = 300;  const int CPav = 20;  //-----------------------------------------------------------------------  struct Albumai {  ...  };  //-----------------------------------------------------------------------  void Skaityti(Albumai A[], int & n);  void Spausdinti(Albumai A[], int n);  void Rikiuoti(Albumai A[], int n);  long Suma(Albumai A[], int n, int p);  //-----------------------------------------------------------------------  int main()  {  Aprašyti masyvą SAlbumai  Skaityti duomenis prieš tai patikrinti ar faile yra duomenys  ........  Rikiuoti masyvą  Spausdinti rezultatą  }  return 0;  }  //-----------------------------------------------------------------------  void Skaityti(Albumai A[], int & n)  {  ...  }  //-----------------------------------------------------------------------  void Spausdinti(Albumai A[], int n)  {  ...  }  //-----------------------------------------------------------------------  void Rikiuoti(Albumai A[], int n)  {  ...  }  }  //-----------------------------------------------------------------------  long Suma(Albumai A[], int n, int p)  {  int s = 0;  ...  return s;  } |