**Užduotis**

Kauno miestas surengė šventę, kurios metu yra žaidžiami įvairūs žaidimai. Žaidimai yra suskirstyti į kategorijas k (1 ≤ k ≤ 10). Taip pat yra fiksuojamas dalyvių skaičius m (5 ≤ m ≤ 20). Kiekvienas dalyvis pasirenka vieną žaidimo kategoriją ir žaidžia po 5 kartus. Kiekvieno žaidimo metu yra fiksuojami taškai.

Jeigu nesusirenka minimalus kiekis dalyvių ar dalyvis pasirenka neegzistuojančią žaidimo kategoriją, žaidimas yra nevertinamas.

Parašykite programą, kuri iš surinktų duomenų suskaičiuotų, kiek iš viso buvo pelnyta taškų kiekvienoje žaidimų kategorijoje.

**Pradiniai duomenys**

Pirmoje eilutėje yra pateikiamas dalyvių skaičius M.

Tolesnėse eilutėse nurodoma kiekvieno dalyvio pasirinkta kategorija K bei surinktas taškų skaičius per visus 5 žaidimus.

Duomenys gali būti nuskaitomi iš tekstinio failo arba konsolės (pasirinkti vieną iš būdų).

**Pavyzdys:**

**Pradiniai duomenys:**

|  |  |
| --- | --- |
| 10 | M – dalyvių skaičius |
| 1 15 20 22 5 9 | K – žaidimo kategorija, surinkti taškai |
| 5 0 6 12 14 5 | K – žaidimo kategorija, surinkti taškai |
| 2 12 12 5 14 13 | K – žaidimo kategorija, surinkti taškai |
| 7 1 1 9 6 7 | K – žaidimo kategorija, surinkti taškai |
| 8 11 17 15 13 9 | K – žaidimo kategorija, surinkti taškai |
| 10 12 20 13 14 16 | K – žaidimo kategorija, surinkti taškai |
| 2 11 12 11 14 9 | K – žaidimo kategorija, surinkti taškai |
| 4 12 16 14 12 16 | K – žaidimo kategorija, surinkti taškai |
| 1 9 6 8 7 12 | K – žaidimo kategorija, surinkti taškai |
| 9 11 12 11 15 16 | K – žaidimo kategorija, surinkti taškai |

**Gautas rezultatas:**

|  |  |
| --- | --- |
| 1 113 | Žaidimo kategorija, surinkti taškai |
| 2 113 | Žaidimo kategorija, surinkti taškai |
| 3 0 | Žaidimo kategorija, surinkti taškai |
| 4 70 | Žaidimo kategorija, surinkti taškai |
| 5 37 | Žaidimo kategorija, surinkti taškai |
| 6 0 | Žaidimo kategorija, surinkti taškai |
| 7 24 | Žaidimo kategorija, surinkti taškai |
| 8 65 | Žaidimo kategorija, surinkti taškai |
| 9 65 | Žaidimo kategorija, surinkti taškai |
| 10 75 | Žaidimo kategorija, surinkti taškai |

**Duomenys**

**1 variantas**

|  |  |
| --- | --- |
| 11 | M – dalyvių skaičius |
| 3 11 25 6 14 3 |  |
| 5 20 20 15 4 9 | K – žaidimo kategorija, surinkti taškai |
| 8 10 16 1 4 15 | K – žaidimo kategorija, surinkti taškai |
| 3 11 2 15 4 1 | K – žaidimo kategorija, surinkti taškai |
| 1 11 12 8 12 17 | K – žaidimo kategorija, surinkti taškai |
| 9 10 7 3 3 8 | K – žaidimo kategorija, surinkti taškai |
| 9 2 2 3 4 6 | K – žaidimo kategorija, surinkti taškai |
| 10 13 22 7 4 19 | K – žaidimo kategorija, surinkti taškai |
| 3 2 6 4 2 6 | K – žaidimo kategorija, surinkti taškai |
| 2 5 12 9 7 2 | K – žaidimo kategorija, surinkti taškai |
| 1 12 12 12 15 17 | K – žaidimo kategorija, surinkti taškai |

**2 variantas**

|  |  |
| --- | --- |
| 3 | M – dalyvių skaičius |
| 1 1 2 2 3 8 | K – žaidimo kategorija, surinkti taškai |
| 5 10 12 13 5 5 | K – žaidimo kategorija, surinkti taškai |
| 2 10 10 15 4 3 | K – žaidimo kategorija, surinkti taškai |

**3 variantas**

|  |  |
| --- | --- |
| 9 | M – dalyvių skaičius |
| 1 15 20 22 5 9 | K – žaidimo kategorija, surinkti taškai |
| 3 0 6 12 14 5 | K – žaidimo kategorija, surinkti taškai |
| 5 2 2 3 4 6 | K – žaidimo kategorija, surinkti taškai |
| 7 1 1 9 6 7 | K – žaidimo kategorija, surinkti taškai |
| 9 10 10 15 4 3 | K – žaidimo kategorija, surinkti taškai |
| 2 12 20 13 14 16 | K – žaidimo kategorija, surinkti taškai |
| 4 11 12 11 14 9 | K – žaidimo kategorija, surinkti taškai |
| 6 1 2 2 3 8 | K – žaidimo kategorija, surinkti taškai |
| 8 11 2 15 4 1 | K – žaidimo kategorija, surinkti taškai |

**4 variantas**

|  |  |
| --- | --- |
| 5 | M – dalyvių skaičius |
| 1 0 6 12 14 5 | K – žaidimo kategorija, surinkti taškai |
| 2 1 2 2 3 8 | K – žaidimo kategorija, surinkti taškai |
| 12 13 22 7 4 19 | K – žaidimo kategorija, surinkti taškai |
| 4 1 1 9 6 5 | K – žaidimo kategorija, surinkti taškai |
| 5 12 12 5 14 13 | K – žaidimo kategorija, surinkti taškai |

**Papildoma užduotis** (Užduotis nėra privaloma)

Programa turėtų surasti žaidimo kategoriją, kuri surinko daugiausiai taškų, ir kategoriją, kuri surinko mažiausiai taškų. Žaidimo kategorijos, kurios nebuvo žaistos, atrinkime nedalyvauja.

**Rezultatai**

**Gauti rezultatai**

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Programinis kodas**

|  |  |
| --- | --- |
| using System;  using System.IO;  using System.Linq;  using System.Collections.Generic;  class Program  {  static void Main()  {  const int CASE\_COUNT = 4; //Duomenu variantų kiekis  for (int caseNo = 1; caseNo <= CASE\_COUNT; caseNo++)  {  Console.WriteLine($"=== Variantas {caseNo} ===");  string fileName = $"variantas{caseNo}.txt"; //failo pavadinimas  string path = Path.Combine(Directory.GetCurrentDirectory(), fileName); //pilnas kelias iki failo  if (!File.Exists(path))  {  Console.WriteLine($"Failas {fileName} nerastas\n");  continue;  }  var lines = File.ReadAllLines(path);  if (lines.Length == 0)  {  Console.WriteLine("Tuščias failas.\n");  continue;  }  int m = int.Parse(lines[0]); //Nuskaito pirmą eilutę, kuri nurodo žaidėjų kiekį    Console.WriteLine("Pradiniai duomenys:");  System.Console.WriteLine(" " + m);  for (int i = 1; i <= m; i++)  Console.WriteLine($" {lines[i]}");  Console.WriteLine();  if (m < 5 || m > 20)  {  Console.WriteLine($"Žaidėjų kiekis turi būti tarp 5 ir 20. Dabartinis Žaidėjų kiekis: {m}\n");  continue;  }  var totalByCategory = new int[m];  var seenCategories = new List<(int category, int sum)>();  for (int i = 0; i < m; i++)  {  var parts = lines[i + 1]  .Split(' ', StringSplitOptions.RemoveEmptyEntries); //Padalina eilutę į dalis  int k = int.Parse(parts[0]); //Pirmas skaičius yra kategorija  int[] sc = parts.Skip(1)  .Take(5)  .Select(int.Parse)  .ToArray(); //Kiti penki skaičiai yra žaidėjo taškai  if (k < 1 || k > 10)  {  Console.WriteLine($"Kategorija {k} yra netinkama. Žaidimas nevertinamas.");  goto NextCase;  }  totalByCategory[k - 1] += sc.Sum(); //Pridedame žaidėjo taškus prie atitinkamos kategorijos  }  Console.WriteLine("Kategorijų taškai:");  for (int cat = 0; cat < totalByCategory.Length; cat++)  {  int sum = totalByCategory[cat];  Console.WriteLine($" {cat + 1}: {sum}");  if (sum > 0)  seenCategories.Add((cat + 1, sum)); //Naudojame tik tas kategorijas, kurių taškų kiekis > 0  }  MinMax(seenCategories, out var min, out var max);  Console.WriteLine($"\nMažiausiai taškų surinko {min.category} kategorija ({min.sum} taškai).");  Console.WriteLine($"Daugiausiai taškų surinko {max.category} kategorija ({max.sum} taškai).");  NextCase:  Console.WriteLine();  }  }  /// <summary>  /// Apskaičiuoja mažiausią ir didžiausią taškų sumą iš pateiktų kategorijų.  /// </summary>  static void MinMax(  List<(int category, int sum)> scores,  out (int category, int sum) min,  out (int category, int sum) max  ){  min = max = scores[0]; //Pradžioje prilyginame min ir max pirmajam sarašo elementui  for (int i = 1; i < scores.Count; i++)  {  if (scores[i].sum < min.sum) min = scores[i];  if (scores[i].sum > max.sum) max = scores[i];  }  }  } |  |