**A**

*Az alábbi feladatok megoldásához az előadáson bevezetett osztálykönyvtárat kell használnia, azt nem módosíthatja. Az osztály-sablonok kódja megtalálható a* [*http://people.inf.elte.hu/gt/prog/library.zip*](http://people.inf.elte.hu/gt/prog/library.zip) *állományban. A megoldásokat az előadáson látott módon tevékenység objektumokkal kell megvalósítani, amelyeknek osztálya vagy az öt programozási tétel (Summation, Counting, Selection, LinSearch, MaxSearch) osztálysablonjának valamelyikéből származik, vagy az általános felsoroló (Enumerator) osztálysablonból. Nem definiálhatja felül a run(), body(), loopCond() metódusokat, az init()-et is csak akkor, ha a Summation osztályból származtat! A saját kódban egy ifstream típusú objektum >> operátora csak az Enumerator osztályból származtatott osztály first() illetve next() metódusaiban használható. A saját kódban egyáltalán ne szerepeljen ciklus, illetve rekurzív függvényhívás! Konzolra vagy szöveges állományba (általában ostream-re) írni vagy a főprogramban, vagy a Summation result változójának segítségével lehet. A bemeneti adatokat tartalmazó szöveges állományokról feltételezzük, hogy helyesen vannak kitöltve, csak a létezésüket kell ellenőrizni. A szöveges állományt csak egyszer nyithatja meg olvasásra és nem használhat a szöveges állomány sorainak számától függő méretű változót. Döntse el, hogy a feladata üres fájlra is értelmezhető-e: ha nem, kezelje le külön ezt az esetet!*

Egy vidámpark céllövöldéiben feljegyezték, hogy melyik vendég milyen nyereménnyel távozott egy sorozat lövés után. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy céllövölde azonosítóját (string) találjuk, majd a vendég nevét (string), azt követően a nyereményét (string), amely lehet speciálisan az is, hogy „nem\_nyert”. A szöveges állomány céllövöldék szerint, azon belül vendégek szerint növekedően rendezett. Egy soron belül az adatokat szóközök vagy tabulátor jelek választják el.

Példa az állomány néhány sorára:

Chicago Lajos plüss\_maci

Chicago Lajos plüss\_maci

Western Béla nem\_nyert

Western Béla elemes\_manguszta

Western Lajos plüss\_maci

***Megfelelt szintű*** (közepes) a munkája akkor, ha megad egy olyan céllövöldét, amelyikben a legtöbb „nem\_nyert” eredmény született.

***Kiváló szintű*** (jeles) a munkája akkor, ha a megfelelt szinten túl azt is megadja, hogy volt-e olyan céllövölde, ahol senki sem nyert.

Mivel a feladatok egymásra épülnek, az értékelésnél csak a legtöbb pontot érőt kell bemutatni!

Miután a programját bemutatta és azt elfogadták, töltse fel azt Neptun-kód.zip formában: Windows-on a [\\nas1.inf.elte.hu\ZH1\Programozas](file:///\\nas1.inf.elte.hu\ZH1\Programozas), Linux-on **smb://nas1.inf.elte.hu/ZH1/Programozas** könyvtárba.

**B**

*Az alábbi feladatok megoldásához az előadáson bevezetett osztálykönyvtárat kell használnia, azt nem módosíthatja. Az osztály-sablonok kódja megtalálható a* [*http://people.inf.elte.hu/gt/prog/library.zip*](http://people.inf.elte.hu/gt/oep/library.zip) *állományban. A megoldásokat az előadáson látott módon tevékenység objektumokkal kell megvalósítani, amelyeknek osztálya vagy az öt programozási tétel (Summation, Counting, Selection, LinSearch, MaxSearch) osztálysablonjának valamelyikéből származik, vagy az általános felsoroló (Enumerator) osztálysablonból. Nem definiálhatja felül a run(), body(), loopCond() metódusokat, az init()-et is csak akkor, ha a Summation osztályból származtat! A saját kódban egy ifstream típusú objektum >> operátora csak az Enumerator osztályból származtatott osztály first() illetve next() metódusaiban használható. A saját kódban egyáltalán ne szerepeljen ciklus, illetve rekurzív függvényhívás! Konzolra vagy szöveges állományba (általában ostream-re) írni vagy a főprogramban, vagy a Summation result változójának segítségével lehet. A bemeneti adatokat tartalmazó szöveges állományokról feltételezzük, hogy helyesen vannak kitöltve, csak a létezésüket kell ellenőrizni. A szöveges állományt csak egyszer nyithatja meg olvasásra és nem használhat a szöveges állomány sorainak számától függő méretű változót. Döntse el, hogy a feladata üres fájlra is értelmezhető-e: ha nem, kezelje le külön ezt az esetet!*

Egy vidámpark céllövöldéiben feljegyezték, hogy melyik vendég milyen nyereménnyel távozott egy sorozat lövés után. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy céllövölde azonosítóját (string) találjuk, majd a vendég nevét (string), azt követően a nyereményét (string), amely lehet speciálisan az is, hogy „nem\_nyert”. A szöveges állomány vendégek szerint, azon belül nyeremények szerint növekedően rendezett. Egy soron belül az adatokat szóközök vagy tabulátor jelek választják el.

Példa az állomány néhány sorára:

Western Béla nem\_nyert

Western Béla elemes\_manguszta

Chicago Lajos plüss\_maci

Chicago Lajos plüss\_maci

Western Lajos plüss\_maci

***Megfelelt szintű*** (közepes) a munkája akkor, ha megad egy olyan vendéget, aki a legtöbb nyereményt gyűjtötte be.

***Kiváló szintű*** (jeles) a munkája akkor, ha a megfelelt szinten túl azt is megadja, hogy vajon minden vendég nyert-e plüss macit.

Mivel a feladatok egymásra épülnek, az értékelésnél csak a legtöbb pontot érőt kell bemutatni!

Miután a programját bemutatta és azt elfogadták, töltse fel azt Neptun-kód.zip formában: Windows-on a [\\nas1.inf.elte.hu\ZH1\Programozas](file:///\\nas1.inf.elte.hu\ZH1\Programozas), Linux-on **smb://nas1.inf.elte.hu/ZH1/Programozas** könyvtárba.

**C**

*Az alábbi feladatok megoldásához az előadáson bevezetett osztálykönyvtárat kell használnia, azt nem módosíthatja. Az osztály-sablonok kódja megtalálható a* [*http://people.inf.elte.hu/gt/prog/library.zip*](http://people.inf.elte.hu/gt/oep/library.zip) *állományban. A megoldásokat az előadáson látott módon tevékenység objektumokkal kell megvalósítani, amelyeknek osztálya vagy az öt programozási tétel (Summation, Counting, Selection, LinSearch, MaxSearch) osztálysablonjának valamelyikéből származik, vagy az általános felsoroló (Enumerator) osztálysablonból. Nem definiálhatja felül a run(), body(), loopCond() metódusokat, az init()-et is csak akkor, ha a Summation osztályból származtat! A saját kódban egy ifstream típusú objektum >> operátora csak az Enumerator osztályból származtatott osztály first() illetve next() metódusaiban használható. A saját kódban egyáltalán ne szerepeljen ciklus, illetve rekurzív függvényhívás! Konzolra vagy szöveges állományba (általában ostream-re) írni vagy a főprogramban, vagy a Summation result változójának segítségével lehet. A bemeneti adatokat tartalmazó szöveges állományokról feltételezzük, hogy helyesen vannak kitöltve, csak a létezésüket kell ellenőrizni. A szöveges állományt csak egyszer nyithatja meg olvasásra és nem használhat a szöveges állomány sorainak számától függő méretű változót. Döntse el, hogy a feladata üres fájlra is értelmezhető-e: ha nem, kezelje le külön ezt az esetet!*

Egy sörfesztivál söntéseiben feljegyezték, hogy melyik vendég milyen sörből mennyit vásárolt. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy söntés azonosítóját (string) találjuk, majd a vendég nevét (string), azt követően a vásárolt sör fajtáját (string) és a mennyiségét (karakter: ’P’ ~ pohár, ’K’ ~ korsó). A szöveges állomány söntések szerint, azon belül vendégek szerint növekedően rendezett. Egy soron belül az adatokat szóközök vagy tabulátor jelek választják el.

Példa az állomány néhány sorára:

Pécsi Béla szüretlen K

Pécsi Lajos pilsner P

Pécsi Lajos barna K

Leffe Kati világos P

***Megfelelt szintű*** (közepes) a munkája akkor, ha megad egy olyan söntést, amelyikben a legtöbb korsó sör fogyott.

***Kiváló szintű*** (jeles) a munkája akkor, ha a megfelelt szinten túl azt is megadja, hogy volt-e olyan söntés, ahol senki sem ivott „barna” sört.

Mivel a feladatok egymásra épülnek, az értékelésnél csak a legtöbb pontot érőt kell bemutatni!

Miután a programját bemutatta és azt elfogadták, töltse fel azt Neptun-kód.zip formában: Windows-on a [\\nas1.inf.elte.hu\ZH1\Programozas](file:///\\nas1.inf.elte.hu\ZH1\Programozas), Linux-on **smb://nas1.inf.elte.hu/ZH1/Programozas** könyvtárba.

**D**

*Az alábbi feladatok megoldásához az előadáson bevezetett osztálykönyvtárat kell használnia, azt nem módosíthatja. Az osztály-sablonok kódja megtalálható a* [*http://people.inf.elte.hu/gt/prog/library.zip*](http://people.inf.elte.hu/gt/oep/library.zip) *állományban. A megoldásokat az előadáson látott módon tevékenység objektumokkal kell megvalósítani, amelyeknek osztálya vagy az öt programozási tétel (Summation, Counting, Selection, LinSearch, MaxSearch) osztálysablonjának valamelyikéből származik, vagy az általános felsoroló (Enumerator) osztálysablonból. Nem definiálhatja felül a run(), body(), loopCond() metódusokat, az init()-et is csak akkor, ha a Summation osztályból származtat! A saját kódban egy ifstream típusú objektum >> operátora csak az Enumerator osztályból származtatott osztály first() illetve next() metódusaiban használható. A saját kódban egyáltalán ne szerepeljen ciklus, illetve rekurzív függvényhívás! Konzolra vagy szöveges állományba (általában ostream-re) írni vagy a főprogramban, vagy a Summation result változójának segítségével lehet. A bemeneti adatokat tartalmazó szöveges állományokról feltételezzük, hogy helyesen vannak kitöltve, csak a létezésüket kell ellenőrizni. A szöveges állományt csak egyszer nyithatja meg olvasásra és nem használhat a szöveges állomány sorainak számától függő méretű változót. Döntse el, hogy a feladata üres fájlra is értelmezhető-e: ha nem, kezelje le külön ezt az esetet!*

Egy sörfesztivál söntéseiben feljegyezték, hogy melyik vendég milyen sörből mennyit vásárolt. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy söntés azonosítóját (string) találjuk, majd a vendég nevét (string), azt követően a vásárolt sör fajtáját (string) és a mennyiségét (karakter: ’P’ ~ pohár, ’K’ ~ korsó). A szöveges állomány vendégek szerint, azon belül söntések szerint növekedően rendezett. Egy soron belül az adatokat szóközök vagy tabulátor jelek választják el.

Példa az állomány néhány sorára:

Pécsi Béla szüretlen K

Leffe Kati világos P

Pécsi Lajos pilsner P

Pécsi Lajos barna K

***Megfelelt szintű*** (közepes) a munkája akkor, ha megad egy olyan vendéget, aki a legtöbb korsóval ivott.

***Kiváló szintű*** (jeles) a munkája akkor, ha a megfelelt szinten túl azt is megadja, hogy vajon minden vendég ivott-e korsóval.

Mivel a feladatok egymásra épülnek, az értékelésnél csak a legtöbb pontot érőt kell bemutatni!

Miután a programját bemutatta és azt elfogadták, töltse fel azt Neptun-kód.zip formában: Windows-on a [\\nas1.inf.elte.hu\ZH1\Programozas](file:///\\nas1.inf.elte.hu\ZH1\Programozas), Linux-on **smb://nas1.inf.elte.hu/ZH1/Programozas** könyvtárba.