

<b>NÉV:</b>	<b>Eredmény:</b>
<b>NEPTUN KÓD:</b>	
<b>GÉPSORSZÁM:</b>	<b>Elfogadó tanár:</b>

## 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> 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 horgászversenyen feljegyezték, hogy az egyes horgászok milyen halakat fogtak. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy horgász azonosítóját (string) majd azt követően a fogásait találjuk. Egy fogás tartalmazza a kifogott hal fajtáját (string), és méretét (int) cm-ben. Egy soron belül az adatokat szóközök vagy tabulátor jelek választják el.

Példa az állomány egy sorára:

Pisti keszeg 30 ponty 25 harcsa 60

**Megfelelt szintű** (közepes) a munkája akkor, ha horgászonként megadja az általuk kifogott legnagyobb hal fajtáját és méretét. Feltehetjük, hogy minden horgász fogott legalább egy halat.

**Kiváló szintű** (jeles) a munkája akkor, ha a megfelelt szinten túl, ha megadja azt is, hogy volt-e olyan horgász, aki legalább öt halat fogott.

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

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

NÉV:	Eredmény:
NEPTUN KÓD:	
GÉPSORSZÁM:	Elfogadó tanár:

## 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> 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 horgászversenyen feljegyezték, hogy az egyes horgászok milyen halakat fogtak. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy horgász azonosítóját (string) majd azt követően a fogásait találjuk. Egy fogás tartalmazza a kifogott hal fajtáját (string), és méretét (int) cm-ben. Egy soron belül az adatokat szóközök vagy tabulátor jelek választják el.

Példa az állomány egy sorára:

```
Pisti keszeg 30 ponty 25 harcsa 60
```

**Megfelelt szintű** (közepes) a munkája akkor, ha megadja, hogy melyik horgász fogta a legtöbb halat.

**Kiváló szintű** (jeles) a munkája akkor, ha a megfelelt szinten túl megadja azt is, hogy hány horgász fogott pontyot.

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

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

<b>NÉV:</b>	<b>Eredmény:</b>
<b>NEPTUN KÓD:</b>	
<b>GÉPSORSZÁM:</b>	<b>Elfogadó tanár:</b>

## 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> 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 vadászaton feljegyezték, hogy az egyes vadászok milyen vadakat lőttek. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy vadász azonosítóját (string) majd azt követően a zsákmányait találjuk. Egy zsákmány tartalmazza a lelőtt vad fajtáját (string), és súlyát (double) kg-ban. Egy soron belül az adatokat szóközők vagy tabulátor jelek választják el.

Példa az állomány egy sorára:

```
Zsolt róka 8.75 rénszarvas 180.0 emu 130.5
```

**Megfelelt szintű** (közepes) a munkája akkor, ha megadja, hogy van-e olyan vadászt, aki rénszarvast lőtt.

**Kiváló szintű** (jeles) a munkája akkor, ha a megfelelt szinten túl megadja vadászonként azt is, hogy ki hány vadat lőtt.

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

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

<b>NÉV:</b>	<b>Eredmény:</b>
<b>NEPTUN KÓD:</b>	
<b>GÉPSORSZÁM:</b>	<b>Elfogadó tanár:</b>

## 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> 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 vadászaton feljegyezték, hogy az egyes vadászok milyen vadakat lőttek. Az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy vadász azonosítóját (string) majd azt követően a zsákmányait találjuk. Egy zsákmány tartalmazza a lelőtt vad fajtáját (string), és súlyát (double) kg-ban. Egy soron belül az adatokat szóközők vagy tabulátor jelek választják el.

Példa az állomány egy sorára:

```
Zsolt róka 8.75 rénszarvas 180.0 emu 130.5
```

**Megfelelt szintű** (közepes) a munkája akkor, ha megadja, hogy hány vadász lőtt kizárólag rénszarvasokat.

**Kiváló szintű** (jeles) a munkája akkor, ha megfelelt szinten túl megmondja, hogy van-e olyan vadászt is, aki összeségében 1000 kg-nál több vadat lőtt.

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

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

NÉV:	Eredmény:
NEPTUN KÓD:	
GÉPSORSZÁM:	Elfogadó tanár:

## E

*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/oaf/lib.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 receptkönyvben megadják, hogy az egyes ételekhez milyen hozzávalók kellenek. Ezeket az adatokat egy szöveges állományban rögzítették úgy, hogy minden sorban egy recept azonosítóját (string) majd azt követően a hozzávalóit találjuk. Egy hozzávalónak van neve (string), mennyisége (double), és mértékegysége (string). Egy soron belül az adatokat szóközök vagy tabulátor jelek választják el.

Példa az állomány egy sorára:

```
tejbegríz tej 1 liter búzadara 13 evőkanál vaj 6 dkg cukor 5 evőkanál
```

**Megfelelt szintű** (közepes) a munkája akkor, ha kiválogatja azokat az ételeket, amelyekhez cukorra van szükség.

**Kiváló szintű** (jeles) a munkája akkor, ha a megfelelt szinten túl megad egy olyan ételt is, amelyhez a legtöbb összetevő kell.

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

Miután a programját bemutatta és azt elfogadták, tölts fel azt Neptun-kód.zip formában: Windows-on a <\\nas1.inf.elte.hu\ZH1\Programozas>, Linux-on `smb://nas1.inf.elte.hu/ZH1/Programozas` könyvtárba.