**NÉV: JEGY:**

**Neptun-kód: elfogadó tanár:**

**számítógép száma:**

Az alábbi feladatok megoldásához az előadáson bevezetett osztálykönyvtárat kell használnia. 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(), Do(), LoopCond() metódusokat, az Init()-et is csak akkor, ha a Summation osztályból származtat! A saját kódban nem szerepelhet ifstream típusú objektum, helyette használja a szekvenciális inputfájl felsoroló osztály-sablonját (SeqInFileEnumerator), és kezelje le annak OPEN\_ERROR kivételét! A saját kódban egyáltalán ne szerepeljen ciklus. Nem használhat rekurzív függvényhívást! A bemeneti adatokat tartalmazó szöveges állományokról feltételezzük, hogy helyesen vannak kitöltve, csak a létezésüket kell ellenőrizni.

***1.Feladat. (2-esért)*** Két szöveges állományban vállalkozóknak meghirdetett munkák adatait találjuk. Minden sorban egyetlen munka adatai vannak: a munka neve (sztring), befektetési költsége (egész szám) és a várható haszon (egész szám) szóközökkel elválasztva. Építsünk fel külön-külön mindkét fájlból egy-egy egyirányú fejelemes listát úgy, hogy a listaelemek az adott fájlbeli adatokat tartalmazzák, majd írjuk is ki a láncolt listákban tárolt adatokat a képernyőre!

***2.Feladat. (3-asért)*** Az előző feladat megoldásában felépített egyik listában keressük meg az első olyan munkát, ahol a haszon kisebb a befektetésnél, és írjuk ki ennek az adatait is a képernyőre.

***3.Feladat.*** ***(4-esért)*** Az előző feladat megoldásában fűzzük is ki az első olyan munkát tartalmazó listaelemet a listából, ahol a haszon kisebb a befektetésnél, majd írjuk ki az így módosított láncolt lista adatait is a képernyőre!

***4.Feladat. (5-ösért)*** Az eddigieken túl listázzuk ki a képernyőre a két láncolt lista összefuttatásából származó adatokat, de ha ugyanaz a nevű munka mindkét állományban szerepelt, akkor a munka neve mellé a nagyobbik befektetési költséget és a kisebbik hasznot írjuk ki! Feltehetjük, hogy mindkét állomány munka nevek szerint szigorúan növekvően rendezett.

***Tippek:***

1.Feladathoz: Egy láncolt listát felépítő objektum osztályát származtathatja például a Summation< Job, Node\*>-ból (a Job egy munka nevéből, költségéből és hasznából álló rekord, a Node pedig a lista egy elemének típusa) úgy, hogy annak konstruktora létrehozza a lista fejelemét, a Result() metódusa pedig visszaadja a fejelem címét. A kiíráshoz szüksége lesz a láncolt lista tartalmának (Job) felsorolására.

3.Feladathoz: Definiáljon a láncolt listához olyan bejárást is, amely nem a listaelemek tartalmait sorolja fel, hanem a címeit. Pontosabban cím párokat, azaz az aktuális listaelem címével együtt az azt közvetlen megelőző listaelem címét is visszaadja a felsorolás, mert ez egyszerűvé teszi majd a listából való kifűzést.

Miután a programját bemutatta és azt elfogadták, töltse fel Neptun-kód.zip formában a [\\inf.elte.hu\dfs\zh\OAF](file:///\\inf.elte.hu\dfs\zh\OAF) helyre.