

Programfejlesztés (SZ35MI)

tantárgy

Az 2005/2006 tanév második félévi programozási feladatok

**Beágyazott programozó
Szoftverfejlesztő
Grafikus**

szakirányoknak

A jelen feladatkiírás az általános szabállyal (PRFALT.DOC) együtt értelmezendő !

1. Munkaszervezés

1.1. Feladatválasztás

A feladatot kiscsoportos formában kell vállalni és elkészíteni. Egy kiscsoport lehet 1, 2 vagy 3 tagú. A feladatot az alábbiakban közölt feladatokból kell kiválasztani úgy, hogy :

ugyanazzal a feladattal, ugyanazon fejlesztő környezetben legfeljebb 6 ember foglalkozhat

A programot és a dokumentációt csoportonként külön - külön kell elkészíteni. A csoporton belüli munkamegosztástól függetlenül a csoport minden tagja egyénileg, a teljes feladatért felelős, a teljes feladatból számoltatható be.

A feladatválasztás jelentkezési sorrendben történik.

1.2. Feladatbeadás, értékelés

A szigorlatra bocsátás feltétele a félévi feladat elfogadható szintű előzetes teljesítése. A teljesítés véghatárideje a szorgalmi időszak vége.

A feladat értékelése a jelen dokumentumhoz tartozó szigorlati feladat értékelőlap szerint történik. Ha a feladat valamely értékelési szempontból nem megfelelő, kijavításra visszaadjuk. A feladat anyagára és a megvalósítás módjára vonatkozó esetenkénti ellenőrző kérdések a szigorlati szóbeli részét képezik.

Külön is kiemeljük, hogy a kitűzött feladatokban lényeges szerepet tölt be a grafika. Törekedjünk arra, hogy az információkezelés minél nagyobb mértékben grafikus legyen. Nem

Programfejlesztés (sz35mi)

kizárt természetesen a szöveges adatkezelés sem, ahol ez indokolható.

A feladatot (dokumentáció és programlemez) legkésőbb az utolsó konzultáción be kell adni!

A feladatra maximum 20 pont adható. A ponthatárok a következők:

0 – 9p.	elégtelen
10 – 12p.	elégséges
13 – 15p.	közepes
16 – 18p.	jó
19 – 20p.	jeles

A határidő után teljesített feladatot csak csökkent értékben számítjuk be a szigorlati jegybe!

0 – 9p.	elégtelen
10 – 15p.	elégséges
16 – 20p.	közepes

2. Specialitások

Az alábbiakban az *általános szabályzathoz hozzáveendő* aktuális előírásokat adjuk meg.

- **Program**

A programot Turbo C 2.01 környezetben kell fejleszteni. A program grafikus fejlesztéséhez javasolt a **konzolio.prj** projektben használt billentyűzetkezelő függvények használata. Az egér- és grafikus képernyő kezelését az előzetesen ismertetett, **EGER**, **GRAFIKA** közös könyvtárban található forrás programok alapján javasolt megoldani. A feladatot **saját fejfájl** (header) használatával kell elkészíteni, ami tartalmazza a feladathoz szükséges adatszerkezeteket és függvényeket. A féléves feladat végterméke minden esetben egy **projekt fájl legyen!**

- **Dokumentáció**

A *mellékletben* legalább *három* a tesztadatsor, tesztfuttatási eredmény szerepeljen!

3. Feladatok

01. Útnyilvántartás készítése

Fájlban tárolt útvonal adatokból generáljunk lehetséges útvonal listát:

- adott km adatra
- adott fogyasztás adatra, amelyet Ft-ban rögzítünk.

Részfeladatok:

- Útvonal adatok karbantartása (felvitel, módosítás, törlés)
- Alapadatok karbantartása (felvitel, módosítás, törlés): Gépjárművek típusa, fogyasztása, fogyasztott üzemanyag típusa, üzemanyag ára.
- Útvonalak kiválasztása minimális maradék számításával.
- Lista megjelenítése képernyőn.
- Lista nyomtatása LPT1 kimenetre.
- Grafikus szimuláció megvalósítása, ahol:
- A városokat az útnyilvántartásnak megfelelő, generált gráf pontjai, az utakat pedig ezek élei jelentik. A gépjárművet a rajzzal arányos, téglalap szimbolizálja, amelyet végig kell vezetni a generált útnyilvántartásnak megfelelően a gráfon.

02. Ping-pong szimuláció sokszögek segítségével

Grafikus felületen a felhasználó által pontokból kijelölt sokszöget rajzolunk ki.

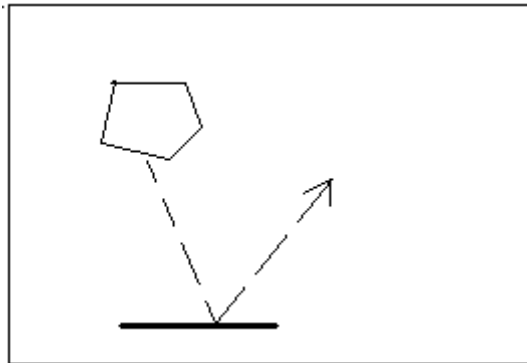
Általános funkciók:

- Új sokszög előállítása a speciális jellemzők előírásának lehetőségével:
 - Véletlenszerű generálással.
 - Felhasználói (a program által kontrollált) megrajzolással. A felhasználó kijelöli a sokszög koordinátáit egér segítségével.
- Sokszög grafikus kijelölése, a kijelölt sokszög grafikus kiemelése. A teljes megoldás az, hogy a sokszög bármely pontjával kijelölhető. Elfogadható gyengébb megoldás, ha a kijelölő pontnak a sokszög kerületére, vagy csúcspontjára kell esnie.
- Sokszög kimentése azonosítóval.
- Sokszög betöltése.
- Kijelölt sokszög elemi transzformációi:
 - Átméretezés

Speciális funkció:

- Mozgatás a játéknak megfelelően: A képernyőn véletlenszerű irányba mozgatott sokszöget, a képernyő alján lévő asztallal kell megakadályozni, hogy leessen. Az asztalt a felhasználó mozgatja jobbra, vagy balra. A sokszög erről (amennyiben sikerül alá csúsztatni a sokszögnek az asztalt) visszapattan felfele, majd folytatja a mozgást. A sokszög a képernyő oldaláról és felső szélétől is visszapattan.

- Alapadatok karbantartása a játékhoz: esés sebessége.
- Játékidő mérése az egyes szimulációk alatt, ezek elmentése azonosítóval, rangsor készítése.



02A. Ping-pong szimuláció háromszög segítségével

- Háromszögek

A kezelendő sokszögek háromszögek. A speciális jellemzők: egyenlő oldalú, egyenlő szárú, derékszögű, hegyesszögű, tompaszögű.

02B. Ping-pong szimuláció négyszög segítségével

- Négyszögek

A kezelendő sokszögek konvex négyszögek. A speciális jellemzők: paralelogramma, trapéz, rombusz, téglalap, négyzet, deltoid, húrnégyszög, érintőnégyyszög.

03. Hanoi tornyai szimuláció

Hanoi tornyai logikai játék. Alapadatok, 3 rúd, páratlan számú korong. Célja, hogy a korongokat egy másik rúdra tegyük át. Szabályok: egyszerre csak egy korong mozgatható, és csak kisebb korong kerülhet egy nagyobbra.

Részfeladatok:

- Menüvezérelt az induló adatok kiválasztása: korongok száma.
- Automatikus végrehajtás: a program lépésenként szemlélteti a végrehajtást gyors sebességgel.
- Manuális végrehajtás: a felhasználó egyenként mozgatja a korongokat, először megjelöli a korongot, amit mozgat, utána az új helyét, ha a lépés ellenkezik a szabályokkal, a program figyelmeztető üzenetet küld.
- Játékidő mérése manuális végrehajtás alatt.
- Egy játék kimentése azonosítóval.
- Játék visszatöltése, az idő nem nullázható.
- Rangsor készítése.

04. Csődarabolást optimalizáló program

Egy vállalatnál 3-5 féle méretű csődarabokra van szükség bukórácsok gyártásához. Az alapanyag mérete mindig azonos. Megoldandó probléma, hogy a lehető legkevesebb hulladékkal előállítsuk a darabokat, annak ismeretében, hogy melyik méretből hány darab kell.

Részfeladatok:

- Csődarabok adatainak karbantartása: **azonosító** egy sorszám, **méret** cm-ben
- Menüvezérelt az induló adatok kiválasztása: **csőméret** (1,2,3,4,5), **darabszám**
- Automatikus végrehajtás
- Egy darabolás kimentése azonosítóval.
- Darabolás visszatöltése.
- Grafikus megjelenítés: szimuláljuk a rudakat téglalapokkal, rajzoljuk ki a darabolás előtti, és utáni állapotot. Késleltetéssel jelenítsük meg a köztes fázisokat is. A késleltetés sebessége legyen változtatható.

05. Elektronikus áramkör készítő, tesztelő program (mini Wokbench)

Alap áramköri elemekből grafikus "tervezőasztalon" egyszerű áramköröket készíthetünk. Az alapelemek menüből választhatóak, és helyének egérrel való kijelölésével helyezzük el az asztalon. Az egyes áramköri elemek paramétereit a tervezőasztalra helyezés előtt rögzíteni kell.

Pl. izzó 6, 12V stb.

A program képes legyen valamely egérrel kijelölt elemre számolt adatokat közölni, pl. hány V feszültség folyik át rajta.

Részfeladatok:

- Áramköri elemek grafikus megjelenítése egyszerű alakzatokkal. Az elemeknek közelíteni kell a szabványos jelöléshez.
- Elemek paramétereinek változtathatónak kell lenni. Ezeket az információkat a grafikus jel mellett meg kell jeleníteni.
- Adatok számolása : áramerősség, ellenállás.
- Helyesség ellenőrzése: Az áramkör nem szakadt-e, működőképes-e.

05A. Izzó

Alkalmazható áramköri elemek:

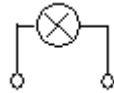
Izzó ⊗

4,5 V elem | |

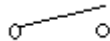
vezeték —

csatlakozási pont •

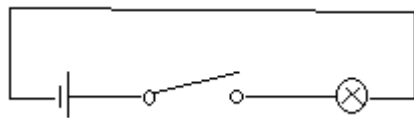
párhuzamosan kötött izzó



kapcsoló



példa:



4,5 V 4 V izzó, 2 W teljesítmény

Számolandó: **áramerősség** az izzóra: $P=U \cdot I$ képletből, $I=P/U=2W/4V=2A$

Ellenállás az izzóra: $R=U/I=2W/2A=1\Omega$

Ugyanezen értékeket több sorosan kötött izzóra, és párhuzamos kötésnél is ki kell tudni számolni.


05B. Ellenállás

Alkalmazható áramköri elemek:

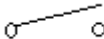
Ellenállás 

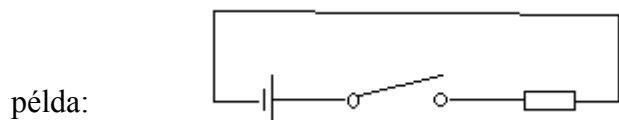
4,5 V elem 

vezeték 

csatlakozási pont 

párhuzamosan kötött ellenállás 

kapcsoló 



4,5 V 2Ω

Számolandó: **áramerősség** : $I=U/R=4,5V/2\Omega=2,25A$

teljesítmény : $P=U*I=4,5V*2,25A=10,125W$

Ugyanezen értékeket több sorosan kötött ellenállásra, és párhuzamos kötésnél is ki kell tudni számolni.

06. Paint jellegű rajzolóprogram

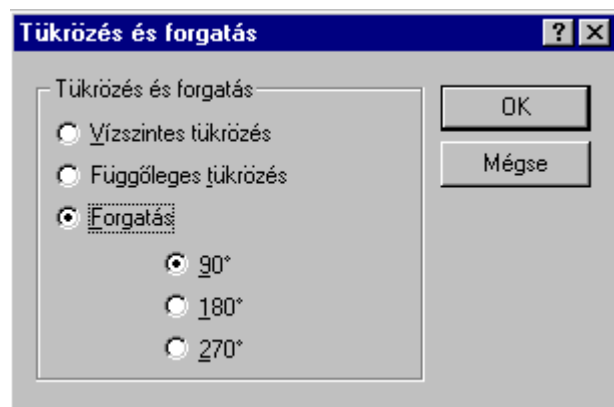
Egyszerű grafikus elemekből "rajzasztalon" ábrát készítünk. Hasonlóan a Paint programhoz, az elemek lehetnek:

Egyenes, görbe vonal, téglalap, kör, ellipszis

Az elemeket egér segítségével lehet felvinni a rajzasztalra.

Részfeladatok:

- Alakzatok grafikus megjelenítése, rajzolása.
- Ábra kimentése azonosítóval
- Elmentett ábra betöltése
- Transzformációk menüből választhatóan:
- Tükrözés-forgatás (egy elemet)
- Nagyítás
- Ábra nyomtatása



07. Kis-keresztrejtvény generálása

Generálandó 2 vízszintes, és 3 függőleges szó, a meghatározásaikkal együtt. A szavakat a betűszámoknak megfelelően négyszögekkel jelöljük a képernyőn. A meghatározásokat jobb oldalt, illetve felül kell elhelyezni. Meghatározás lehet pl. tárgy, fogalom, élőlény stb.

A szavakat fájlban tároljuk a meghatározásaikkal együtt.

Részfeladatok:

- Rejtvény generálás, kirajzolás.
- Egy rejtvény kimentése azonosítóval.
- Rejtvény visszatöltése.
- Input mezőben a lehetséges tipp megadása a következő módon: először egérrel kijelöljük melyik elemet fejtjük éppen, azután beírjuk a megfejtést. Ellenőrizendő, hogy csak azonos hosszúságú szót fogadjunk el! Ha eltaláltuk, a képernyőn is ki kell takarni a megfejtést az adott szóra.
- Menüvezérelt végrehajtás

08. Fájl struktúra szemléltetés

Alapesetben tekintsük a következő egyszerű fájlkezelési feladatot.

Egy szövegfájlban neveket tartunk nyilván, egy név egy rekordot (sort) alkot. A neveket bevitel előtt ellenőrizni kell. A név két részből áll, a vezetéknev nagybetűvel kezdődik, utána kisbetűk állnak, és legalább 2 betű. A keresztnév szintén nagybetűvel kezdődik, utána kisbetűk állnak, és legalább 3 betű. Elválasztó karakter a szóköz. (Pl. Jó Bea)

A fájlműveletek: olvasás a fájlból , írás a fájlba. Az ismert fájlstruktúrának megfelelően:

```
typedef struct {
    short    level; /* a puffer telítettségi/üres szintje */
    unsigned flags; /* fájl állapot flag-ek */
    char     fd; /* fájl leíró */
    unsigned char hold; /* ungetc kar., ha nincs puffer */
    short    bsize; /* puffer mérete */
    unsigned char *buffer; /* adat átviteli puffer */
    unsigned char *curp; /* aktuális fájl mutató */
    unsigned istemp; /* ideiglenes fájl jelző */
    short    token; /* validitás ellenőrzéshez */
} FILE;
```

Jelenítse meg, mi történik a pufferben íráskor, és olvasáskor!

Részfeladatok:

- Menüvezérelt végrehajtás
- Input adat ellenőrzés bevitelkor
- Írás a fájlba
- Olvasás a fájlból
- Puffer tartalmának szemléltetése.

09. Egérmaszk generálás

Készítsen olyan programot, amely megkönnyíti a grafikus egérmutató maszkjának megtervezését. A grafikus felületen négyzetrács segítségével megtervezett maszk adatai legyenek elmenthetőek! Több ilyen fájlból legyen adatcsomag is készíthető, mely tömbökben helyezi el az egyes maszkok adatait!

Részfeladatok:

- Egérmaszk tervezése
- Maszkok kimentése fájlba.
- Menüvezérelt válogatás az adatcsomagokból, betöltés előtt előnézet megmutatása.

- Maszkok betöltése fájlból, a betöltött alkalmazása a grafikus felületen.

10. Függvények ábrázolása

Általános fogalmak és jellemzők

Ebben a feladatcsoportban egy véges intervallumon véges számsorozatokkal egyértelműen definiálható függvények bemutatása és kezelése a feladat. A függvények darabszámát és a definiáló sorozatok elemszámát konstanssal ne korlátozzuk, nyilvánvalóan elfogadható az, hogy az ábrázolás csak bizonyos mérethatárok között lesz vizuálisan is követhető.

Általános fogalmak:

- **Értelmezési tartomány:** A valós számegyenesnek az origóra szimmetrikus, a képernyőn megjeleníthető véges intervalluma, mérete programkonstanssal rögzíthető. Az intervallum pontjai grafikus képernyőpontok (pixelek), a függvényértékek is így számítandók (ha szükséges, kerekítéssel).
- **A függvény grafikonja:** Az értelmezési tartomány pontjaihoz tartozó függvényérték pontok összessége.
- **Függvénycsoport:** egy csoportba tartozó, a képernyőn együtt megjelenő függvények összessége. A csoport elemszáma nem korlátozott, az operatív tárból dinamikus adatszerkezetben, a háttértáron egy fájlban tárolandó. A fájl neve egyben a csoport azonosítója is.

Minden feladatban megvalósítandó általános funkciók:

- Csoport kiválasztása, betöltése, kimentése, megjelenítése, törlése.
- Függvény előállítás véletlenszerű generálással és felvétele a csoportba.
- Függvény előállítás és módosítása felhasználói (a program által kontrollált) grafikus és/vagy szöveges adatmegadással és felvétele a csoportba.
- Függvény grafikus kijelölése.
- Kijelölt függvény törlése a csoportból.
- Kijelölt függvény konstanssal való szorzatának felvétele a csoportba.
- Két kijelölt függvény összegének felvétele a csoportba.
- Két kijelölt függvény különbségének felvétele a csoportba.
- A határozott integrál (grafikon alatti terület) értékének kiszámítása, adott részintervallumon.

10A. Lépcsős függvények

A kezelendő függvények részintervallumonként konstans függvények. Speciális feladat: az integrálfüggvény meghatározása és ábrázolása. Ez nem lépcsős függvény, nem kell felvenni a csoportba.

10B. Törtvonal függvények

A kezelendő függvények részintervallumonként lineáris függvények. Speciális feladat: a derivált függvény meghatározása és ábrázolása. Ez nem törtvonal függvény, nem kell felvenni a csoportba.

10C. Polinom függvények

A kezelendő függvények polinom függvények. Speciális feladatok: a derivált és integrálfüggvény valamint a lineáris transzformált függvény meghatározása és felvétele a csoportba.

Ajánlott irodalom:

*Matron László: Fehérvári Arnold: **Algoritmusok és adatstruktúrák**, Novadat, Győr, 2002.*

*Bauer Péter: **Programozás I-II**. Universitas Kht. Győr, 2005.*

*T.H. Cormen - C.E. Leiserson - R.L. Rivest: **Algoritmusok** Műszaki Könyvkiadó, Budapest*

*A.V. Aho - J.E.Hopcroft - J.D.Ullman: **Számítógép algoritmusok tervezése és analízise** Műszaki Könyvkiadó, Budapest*

Győr 2006. március 13.

Varjasi Norbert
egyetemi tanársegéd