

10. LABORATEGIA

FITXATEGIAK C LENGOAIA

1 HELBURUAK

Jarduera hau amaitzean, ikaslea ondorengo burutzeko gai izango da:

- Fitxategiekin egokiro lan egin.
- Fitxategiak erabiltzen dituzten funtzioak erabili.
- Fitxategi eta hauak maneiatzen dituzten aldagaien arteko ezberdintasuna ulertu.
- Fitxategi eta beraien izenen arteko diferentzia ulertu.
- Fitxategiak atzipen sekuentziala eta zuzeneko atzipena burutuz erabili.
- Fitxategiak erabiltzearen baliagarritasuna egiaztatu.

2 MOTIBAZIOA

- Fitxategiak erabili ezean ebazteko zailak diren problema ugari daude.
- Fitxategian erabilera datuen sarrera eta irteera sinplifikatzen du, behin gorde ezkeror, behar beste alditan erabili bait daitezke.
- Aldaketa txikiekin orain arte egindako programak fitxategiekin erabili daitezke, lortutako emaitzak modu iraunkorrean gordetzea ahalbidetuz. Esate baterako, teklatuaren bitartez bektore bat eskuratzen bada, diskoko fitxategi baten gorde daiteke.
- Bestalde, programa baten lortutako emaitzak diskoko fitxategi baten gordetzen badira, beste edozein programetan erabili daitezke.
- Gainera, okerreko datu bat dagoela baieztatzen bada, modifikatu daiteke gainerako datu guztiak aldatu gabe.
- Datuak zuzenean aldatu daitezke, gero berrerabiltzeko.
- Datuak automatikoki sortu daitezke, gero berrerabiltzeko.
- Emaitzak gorde daitezke, egokiak direla ondoren baieztatzeko.

3 FITXATEGIEN ERABILIERA C-Z

Laborategi honetan erabili daitezkeen funtzioen prototipoak ondorengoak dira:

```
FILE * fopen(const char izena[], const char irekitzea[]);  
int fprintf(FILE * fitxategia, const char formatua[], ...);  
int fputs(const char katea[], FILE *fitxategia);  
int fscanf(FILE * fitxategia, char katea[], ...);  
char * fgets(char katea[], int n, FILE *fitxategia);  
int fclose(FILE *fitxategia);  
int fseek(FILE *fitxategia, long pos, int modua);  
int feof(FILE *fitxategia);  
char * strerror(int errorea);  
int rand(void);  
void srand(unsigned int hazia);
```

4 ARIKETAK

4.1 1. ariketa

4.1.1 1. Helburua

Teklatuaren bitartez sartutako datuak diskoko fitxategi baten gorde.

4.1.2 Enuntziatua

1. Teklatuaren bitartez sartutako datuak "float.txt" fitxategian gordetzea posible egiten duen programa idatzi. Programak zenbaki negativo bat sartzean amaitu behar du.
2. Programa exekutatu.
3. Teklatuaren bitartez sartutako datuak fitxategian gordeta daudenekin bat datorrela egiaztatu.
4. Bigarren aldiz programa exekutatu eta emaitzak baieztatu.
5. Sartutako datuak fitxategiaren bukaeran eransteko programa modifikatu.

Exekuzio adibidea:

(Erabiltzaileak
tekleatutako datuak
*azpimarratuta eta letra
etxanez* agertzen dira)

BALIO ERREAL ETA POSITIBOAK SARTU

X(1): 13.91

X(2): 23.11

X(3): 83.13

X(4): 12.22

X(1): -1

Sakatu tekla bat jarraitzeko . . .

Kasu honetan programaren exekuzio pantailak ez da oso interesgarria emaitzak "float.txt" fitxategian gordeko direlako. Fitxategiak sartu berri diren datuak dituela egiaztatu behar da.

4.1.3 Laguntza

Teklatu bitartez datuak sartzea onartzen duen programa zizkliko bat idatzi, ondoren datu horiek fitxategi berri baten gordetzeko.

Egin beharreko lehenengo gauza, datu sarrerarako programa laburra idaztea da. Erabiltzaileari datu numerikoak eskatu eta zenbaki negativo batekin maitzen duen orain arte egindako edozein programa erabili daiteke.

Programa exekutatu eta gero, sartutako datuek "float.txt" fitxategian gordeta egon beharko dute. Egiaztapena bai fitxategia garapen ingurunean irekiz edo Windows esploratzailearen bitartez egin daiteke.

Gogoratu datuak fitxategi baten gordetzeko ondorengo funtzioa erabili daitekela:

```
int fprintf(FILE * fitxategia, const char formatua[], ...);
```

baina horretarako funtzioa erabiltzean balio bat esleitu ahal zaion fitxategi aldagai bat eduki behar da:

```
FILE * fopen(const char izena[], const char irekitzea[]);
```

Gogoratu kasu honetan irekitze moduak "w" edo "a" izan beharko zuela.

Kontuan hartu irekitze moduaren aldaketaren eragina emaitzak gordeko dituen fitxategian. Problemaren arabera, portaera bat edo bestea komeni izango da.

Interesgarri da baita nola datuak fitxategian modu ezberdinean gorde daitezkeen, erabilitako formatu katearen arabera.

4.2 2. ariketa

4.2.1 Helburua

Programak sortutako datuak diskoan gorde.
Karaktere batzuk irudikapen ikusgaia ez dutela behatu.

4.2.2 Enuntziatua

1. ASCII karaktere multzoa "ascii.txt" fitxategian gordetzen duen programa idatzi.
2. Karaktere multzo luzatua gordetzeko aurreko programa modifikatu.

4.2.3 Laguntza

Kodea 32 baino txikiagoa duten karaktereak pantailan irudikatu ezin direla gogoratu (adibidez, 8 karakterea ez da ikusten eta txistu bat botatzen du).

Fitxategiak ondorengo itxura eduki beharko luke:

```
....  
65 A  
66 B  
67 C  
68 D  
69 E  
76 L  
...
```

Non lehenengo zutabeak bigarrean zutabeko karakterei dagokien balio numerikoa adierazten duen.

4.3 3. ariketa

4.3.1 Helburua

Iadanik datuak fitxategietan nola gordetzen diren ezaguna denez, orain erabili behar dira, hau da, fitxategi baten aurkitzen diren datu numerikoak bere batura eta biderkadura (edo orokorrean beste edozein operazio) ateratzeko erabiliko dira.

4.3.2 Enuntziatua

1. "float.txt"-ko datuak erabiliz, horien batura eta biderkadura lortzen duen programa idatzi. Horretarako, bi funtzio definitu behar dira, bat batuketarako eta beste bat biderketarako, ondoren agertzen diren antzeko prototipoekin:

```
float sum_fich (char * nomfich);  
float mult_fich (char * nomfich);
```

2. "float.txt" fitxategian dezimalak ezabatu ahal direla eta programa ondo dabilela egiaztatu, hau da, datuak bai adierazpide osoan bai dezimalean sartu daitezke.
3. Aurreko programa erabili nahi den fitxategiaren izena teklatuaren bitartez eskuratzeko orokortu.

4.3.3 Laguntza

Aurretik sortutako "float.txt" fitxategia erabili balioak editatzeko edo balio berriak eransteko.

4.4 4. ariketa

4.4.1 Helburua

Ausazko zenbakien sorrera burutzen duen baliabidea erabili fitxategietan gordetzen diren datuak eskuratzeko eta ondoren programen proba betetzeko.

4.4.2 Enuntziatua

1. Erabiltzaileak ematen duen izena duen fitxategi baten, **rand()** funtzioak ausaz sortutako 100 zenbaki gordetzen duen funtzio bat idatzi.
2. Fitxategiaren datuak erabiliz, balio ertaina, bariantza eta desbideratze tipikoa hornitzen duen funtzioa idatzi.
3. Balio oso bat gorde eta "hazia.txt" deitzen den .fitxategia eraiki.
4. Lehenengo puntuak deskribatzen den funtzioa modifikatu zenbaki sekuentzia ezberdinak gordetzeko, "hazia.txt" barruan dagoen balioaren arabera.

4.4.3 Laguntza

- Balioa zuzenean fitxategi batetan gorde daiteke garapen ingurunea erabiliz "hazia.txt" fitxategia zuzenean editatzeko, edo baita ere programa erabiliz hazi berria erabiltzaileari eskatzeko.

4.5 5. ariketa

4.5.1 Helburua

Rand() funtzioan oinarrituta dauden funtzioak idatzi ausazko distribuzio ezberdinak sortzeko eta fitxategiak zuzeneko atzipenaren bidez.

4.5.2 Enuntziatua

1. Makina baten sensoreak adierazten dituen taula bat "sensoreak.txt" fitxategian gordetzen duen funtzioa idatzi, ondoren agertzen den moduan:

```
0001000000
1000000000
0110000000
0000000A00
1000000000
0000000000
0000010000
0100001100
0000000001
0000000000
```

Aurreko taulak 100 sensore dituen makina bat adierazten du (10 lerro eta 10 zutabe). Sensoreen egoera zero eta baten bitartez adierazten da. Zeroak sensorea desaktibatuta dagoela adierazten du eta bata sensorea aktibatuta dagoela adierazten du. Hondatuta dagoen sensorea adierazteko, **A** letra erabiltzen da.

Sensoreen egoeren balioak sortzeko, **matxurak** %1 probabilitatearekin gertatzen direla, sensoreak %10 probabilitatearekin **aktibatzen** direla eta bestelako kasutan **geldi** dagoela koentutan hartu behar da.

Suposatzen da lerro eta zutabe kopuruak konstante ezagunak direla, adibidez LK=10 eta ZK=10:

2. Hondatutako sensore kopurua atera.
3. Fitxategia zuzeneko atzipenean ireki eta 4. lerroa eta 8. zutabeko balioa eskuratu.
4. Aurreko puntua orokortu teklatuaren bitartez sartzen diren edozein lerro eta zutabeari dagokien balioa eskuratzeko, lerro edo zutabe negativo bat sartu arte.
5. "sensoreak.txt" fitxategia automatikoki eguneratu "serie.txt" fitxategia erabiliz. Fitxategi honek lerro bakoitzean sensorearen balioa eta zein lerro eta zutabea dagoen gordetzen du.

4.5.3 Laguntza

- Sensore baten egoera sortzen duen funtzioa idatzi.
- 1 eta 100-ren arteko balioa sortzeko `rand()%100+1` erabili daiteke.
- Lerro bakoitzaren amaieran ikusgaiak ez diren karaktere bi agertzen dira, nahiz eta lerro amaiera karaktere bakarra dagoela pentsatu.