Llenguatges de Programació

Curs 2006 – 07 / Pràctica Final

Avaluació del Bloc 3

Condicions de lliurament:

És MOLT IMPORTANT que llegiu i enteneu bé les condicions de lliurament, per evitar mal entesos el dia de la correcció al laboratori. Penseu que NO S'ACCEPTARÀ CAP PRÀCTICA QUE NO SEGUEIXI LES NORMES que especifiquem a continuació:

1. Heu de fer el lliurament a la sessió de pràctiques del grup on esteu apuntats.
2. TOTS els components del subgrup, hauran de venir el dia de la correcció.
3. La pràctica de cada subgrup, haurà d'estar penjada al campus virtual, abans de les 24:00h. del dia anterior a la correcció (vegeu taula adjunta).
   Donat que només podeu penjar-ho una vegada, aneu amb MOLT de compte de penjar la versió correcta, ja que la versió penjada (i no cap altra) serà la que descarregareu i executareu el dia de la correcció.
   Tingueu en compte que si la vostra pràctica consta de varis fitxers, haureu de penjar un fitxer comprimit `.zip`.
4. La (poca) gent que no tingui accés al Campus Virtual s'haurà de posar en contacte amb el/la vostre/a professor/a de pràctiques, amb uns dies d'antelació al lliurament, per determinar la forma de lliurament.

Nº de grup	**Data Límit de lliurament al Campus Virtual**
Grups 1, 2, 3, 4	Diumenge 21 de gener de 2007 a les 23:59
Grups 6, 7, 8	Dilluns 22 de gener de 2007 a les 23:59
Grups 10, 11	Dimarts 23 de gener de 2007 a les 23:59

1. Introducció

'La Ruleta v.2.0.' no ha estat una excepció i també ha satisfet plenament al propietari del Casino Virtual. És per aquest motiu que us ha encarregat la versió definitiva de La Ruleta. Aquesta versió ja serà totalment operativa i permetrà que el Casino Virtual obri les portes al públic. Recordeu que el programa final haurà d'ajustar-se a les regles de funcionament general del joc que estan explicades a l'enunciat del projecte conjunt d'AP i LP. Per tant, **tots aquells que també feu l'assignatura d'Algorismes i Programació, recordeu que el disseny algorísmic ja l'heu fet a la pràctica d'Algorismes i Programació i per tant el podeu aprofitar!!!**

A continuació us detallarem les principals modificacions que heu de fer a 'La Ruleta v.2.0.' per adaptar-la a la versió final i que bàsicament consistiran en que:

- Es pugui portar un control sobre els beneficis que va generant el Casino.
- Es pugui controlar l'accés de gent no desitjada, ja siguin tramposos o ludòpates.
- Es pugui evitar l'accés de menors d'edat.
- Els jugadors registrats puguin canviar diners per fitxes sempre que vulguin.
- En cada partida només juguin els jugadors que ho desitgin.
- Es puguin fer nous tipus d'apostes: dotzenes, parell/imparell, vermell/negre
- Els jugadors registrats puguin abandonar el Casino quan vulguin.

2. La Ruleta, versió 3.0

2.1 Inici (El Casino obre)

Al iniciar el programa, el primer que haureu de fer, és recuperar informació relacionada amb el Casino i que està guardada en dos fitxers:

- Estat dels comptes del Casino: Del fitxer `ComptesCasino.txt`, llegireu les següents dades:

  1. Data.
  2. Guanys de Comissions.
  3. Guanys d'Apostes Perdudes.
  4. Guanys Totals.
  5. Capital de la banca.

  que guardareu en una estructura de dades apropiada, i la qual anireu actualitzant en temps d'execució. Vegeu el format d'aquest fitxer:

`ComptesCasino.txt`

```
dia
mes
any
GuanysComisions
GuanysApostes
GuanysTotals
CapitalBanca
```

format

```
20
12
2006
137
6233
6470
1550000
```

exemple

  Tingueu en compte que en aquest fitxer només hi ha les dades de l'últim dia. La informació que s'hi guarda és l'acumulació de guanys des del primer dia d'obertura del casino fins al dia anterior. Per tant, haureu de llegir el fitxer al principi del programa, actualitzar els guanys i el capital segons les apostes que es facin durant l'execució i al final tornar a guardar les dades actualitzades amb la nova data.

- Persones no admeses: la llista de persones no admeses, ja sigui per la seva ludopatia o pel fet de fer trampes, figurarà en el fitxer `NoAdmesos.txt`, en el qual hi haurà un primer número que indicarà quants no admesos hi ha al fitxer, i a continuació els seus DNI's respectius. Haureu de pensar quina és l'estructura més apropiada per guardar-los en memòria. Vegeu el format d'aquest fitxer:

`NoAdmesos.txt`

```
N
DNI_1
DNI_2
 .
 .
 .
DNI_N
```

format

```
4
40522439
52335496
45228197
56687216
```

exemple

ATENCIÓ: És IMPRESCINDIBLE que respecteu el nom dels fitxer que us hem indicat, ja que el dia d'entrega, haureu de testejar el funcionament de la pràctica amb els fitxers que us passarem nosaltres, i que tindran aquests noms.

NOTA: Una vegada heu llegit els fitxers, NO us oblideu de tancar-los!

2.2 Funcionament

Una vegada hagueu fet les tasques preliminars (lectura dels fitxers), s'ha de mostrar un menú amb les següents opcions:

1. Registrar jugador.
2. Canviar diners per fitxes.
3. Fer partida.
4. Sortir del Casino.
5. Tancar Casino.

i s'ha de demanar que se'n triï una. Si es tria una opció no vàlida s'ha d'indicar amb un missatge per pantalla. El programa ha de permetre que es vagin triant tantes opcions com es vulgui fins triar la opció 5.

2.2.1 Registrar jugador

Aquesta opció, és essencialment igual que en les versions anteriors, però aquesta vegada haureu de tenir en compte un parell de coses:

- Cal controlar que l'usuari que està entrant al Casino no sigui menor de 18 anys. (Vegeu Apèndix per a la obtenció de la data i hora del sistema)
- Cal controlar que l'usuari que està entrant al Casino no figuri a la llista de No Admesos.
- Cal controlar que l'usuari no estigui ja registrat.

En qualsevol dels tres casos caldrà mostrar un missatge indicant el motiu pel qual no es permet el registre a l'usuari.

2.2.2 Canviar els diners per fitxes

Només podran canviar diners per fitxes aquells jugadors que estiguin registrats. Per tant, cal demanar el DNI a la persona que vol canviar i comprovar que estigui registrat. Si no ho està, cal mostrar un missatge indicant-ho i no se li permetrà fer el canvi.
Per canviar diners per fitxes, cal aplicar la mateixa fórmula de conversió que a la versió 2.0. Una vegada fet el canvi, se li actualitzarà el valor en fitxes que té el jugador a la seva fitxa.

 **RECORDEU:** actualitzar els guanys del casino en concepte de comissions!

2.2.3 Fer partida

Aquesta vegada, quan es triï aquesta opció, mostrareu el següent submenú:

1. Un jugador vol fer una aposta.
2. Girar Ruleta.
3. Tornar al menú principal.

d'aquesta manera, no tots els jugadors hauran de jugar forçosament, sinó només aquells que ho desitgin. Evidentment es podran anar apuntant tants jugadors com es vulgui, fins que es triï l'opció 'Girar Ruleta'. Per tant, per controlar qui és que fa l'aposta, cal demanar el DNI al jugador que vol apostar.

En aquest nova versió, a més de les ja clàssiques apostes: senzilla, doble, quadrada i columna; s'hi afegeixen les fins ara desactivades:

- Vermell / Negre
- Parell / Imparell
- Dotzenes

Si un jugador vol fer una aposta però no té fitxes, no se li ha de permetre fer l'aposta. Tampoc se li ha de permetre fer l'aposta si vol apostar un valor negatiu. Igualment, cap jugador pot quedar-se a números vermells dins la seva fitxa electrònica. Per tant, si un jugador intenta fer una aposta per un import superior al que disposa, se li indicarà amb un missatge que no té fitxes suficients per fer l'aposta i se li mostrarà quin és el valor de les fitxes de què disposa. Per tant, la pregunta de la quantitat que vol apostar es repetirà fins que introdueixi un valor inferior a la quantitat de diners de què disposa.

Recordeu també que en aquesta última versió s'haurà d'implementar la restricció que l'aposta d'un jugador no pot superar el 10% del total del capital de la banca.

Una vegada el jugador ha fet la seva aposta, cal emmagatzemar quins són els números inclosos en l'aposta realitzada i mostrar per pantalla un missatge indicant quins són tots aquests números. També cal restar de la seva fitxa electrònica, la quantitat en fitxes que ha apostat.

Quan es triï l'opció 'Girar Ruleta' s'ha de generar el número sobre el que s'atura la bola de la ruleta i llavors, s'ha de comprovar, per cada jugador, si ha guanyat o ha perdut i s'ha d'actualitzar la seva fitxa electrònica. És a dir, a aquells jugadors que hagin guanyat se'ls ha de sumar la quantitat del premi. A més, mostrarem per pantalla un missatge per cada jugador indicant el seu nom i la quantitat que ha guanyat o ha perdut. Els jugadors que no han apostat, també han de sortir al llistat:

```
>> Jugador XXXX XXXX XXXX ha guanyat XXX euros.
>> Jugador YYYY YYYY YYYY ha perdut YYY euros.
>> Jugador ZZZZ ZZZZ ZZZZ no havia fet aposta.
```

 **RECORDEU:** actualitzar els guanys del casino i el capital de la banca en funció de les pèrdues o guanys dels jugadors!

2.2.4 Sortir del Casino

El fet de triar aquesta opció implica que un determinat jugador vol abandonar el Casino. En aquest cas eliminareu el jugador en qüestió de la llista dinàmica, fareu la conversió automàticament de Fitxes a Diners i mostrareu el següent missatge per pantalla:

```
.....................................................................................................
[hh:mm:ss]
En/na Nom Cognoms, amb DNI Número_DNI i nascut dd/mm/aaaa,
abandona el Casino amb Diners EUR.
.....................................................................................................
```

[`hh:mm:ss`] és l'hora a la que el jugador abandona el Casino. (Vegeu Apèndix per a la obtenció de la data i hora del sistema)

 **NOTA:** No oblideu d'alliberar la memòria que faci falta.

2.2.5 Tancar Casino

Triar aquesta opció indica que la jornada pel Casino ha finalitzat. En aquest cas, haureu de:

- Fer fora als jugadors que queden dins el casino (els que no se'n vulguin anar podrien estar mostrant els primers símptomes de ludopatia), és a dir, buidar la llista dinàmica i mostrar el missatge especificat a l'apartat anterior per a cadascun d'ells.
- Actualitzar el fitxer `ComptesCasino.txt`.

 **NOTA:** No oblideu d'alliberar la memòria i tancar els fitxers que facin falta.

3. Detalls d'implementació

Per implementar la versió final del joc, podeu aprofitar una gran part del codi que teniu de les versions 1.0 i 2.0, així com el pseudocodi que heu fet a la pràctica d'Algorismes i Programació. De tota manera, haureu de fer algunes modificacions en alguna part per adaptar-ho a les especificacions de la versió final.

Tots els consells d'implementació que se us van donar en els enunciats de les pràctiques dels blocs 1 i 2 continuen sent vàlids per aquesta pràctica. Per a realitzar les apostes se us proporcionen algunes funcions en tres fitxers que heu d'incloure en el vostre projecte (RutinesB3.c, RutinesB3.h i MouseUTILS.h). A més també se us proporciona un exemple dels fitxer 'Noadmesos.txt' i 'ComptesCasino.txt', però el dia d'avaluació de la pràctica les proves s'hauran de fer amb els fitxers que us proporcionem aquell dia.

A continuació us detallem alguns punts del funcionament del programa i us expliquem com implementar algunes parts de la pràctica.

Lectura i escriptura dels fitxers

Per fer la lectura i l'escriptura dels fitxers, cal que recordeu:

- Heu d'obrir el fitxer en el mode adequat abans de començar a llegir o escriure.
- Cal comprovar que l'obertura del fitxer s'hagi fet correctament. Si el fitxer no s'ha pogut obrir la funció 'fopen' us retorna NULL. En aquest cas, mostreu un missatge per pantalla i inicialitzeu les estructures de dades on guardeu la informació del fitxer.
- Les lectures i escriptures s'han de fer respectant el format establert i els tipus de dades dels fitxers. Si no llegiu les dades en l'ordre correcte us trobareu que les dades no es carreguen bé. Si no escriviu en el mateix format que se us dóna, les vegades següents que executeu el programa, no podreu llegir el fitxer.
- Cal tancar el fitxer un cop s'ha acabat de llegir o escriure.

Control d'entrada als menors d'edat

Per controlar l'edat de les persones que es volen registrar al Casino, caldrà conèixer la data actual. Per això, tenim funcions del llenguatge que ens permet obtenir la data del rellotge de l'ordinador on s'executa la pràctica. A l'apartat 5 (Annex) se us proporciona la referència d'aquestes funcions i podeu veure un exemple d'utilització.

Estructures de dades

Les estructures de dades que havíeu utilitzat en la versió 2.0 us serviran com a base en aquesta versió final. Ara, però, caldrà definir les estructures de dades necessàries per emmagatzemar la informació sobre els guanys del casino i la llista de jugadors no admesos.

Degut a que en el futur el Casino voldria implementar algorismes optimitzats de cerca sobre la llista de jugadors registrats, cal que la llista estigui ordenada pel camp DNI. Per tant, l'estructura de llista que heu d'implementar és una llista simplement encadenada ordenada pel camp DNI.

IMPORTANT – Recordeu que als apunts de l'assignatura "Algorismes i Programació" teniu el pseudo-codi d'algunes de les funcions necessàries per treballar amb la llista encadenada ordenada. Haureu de fer alguna petita modificació per adaptar-lo a la vostra estructura de dades. Bàsicament, les funcions que us poden ser d'utilitat són:

- **funció** AfegirOrdenat (**var** primer: llista, x: enter): lògic

  que en C i amb les nostres estructures de dades quedaria:

  int AfegirOrdenat (Jugador **primer)

- **funció** EliminarElement (**var** primer: llista, x: enter): lògic

  que en C i amb les nostres estructures de dades quedaria:

  int EliminarElement (Jugador **primer, int DNI)

on Jugador és el tipus definit com a node de la llista.

Registrar jugador

En registrar un jugador, heu de demanar les dades personals i els diners que inicialment vol canviar per fitxes. Segons teniu a l'enunciat del Projecte conjunt d'AP-LP, la fórmula per fer el canvi de diners a fitxes del casino és:

Valor en fitxes = Valor diners x (100 – Percentatge Comissió) / 100

En aquesta versió, definiu el percentatge de comissió com un valor constant ja que se us pot demanar que el canvieu en qualsevol moment. Inicialment, poseu-lo al 5%.

Per comprovar que el jugador que es vulgui registrar, no s'hagi registrat prèviament, caldrà cercar el seu DNI a la llista. Utilitzeu el pseudo-codi de la funció `BuscarElement` que se us proporciona.

Emmagatzemar tota la informació del jugador a la base de dades equivaldrà a fer una inserció de forma ordenada a la llista. Fixeu-vos que ara ja no podeu utilitzar les funcions simples d'inserir pel principi o pel final de la versió 2.0.

Fer partida

En l'opció de fer partida, cada cop que un jugador vol apostar, se li ha de demanar el seu DNI i buscar-lo a la llista per guardar les dades de l'aposta. Fixeu-vos que heu d'utilitzar la mateixa funció per cercar que en l'opció de registrar jugador.

Per fer la comprovació dels resultats de la partida i fer el llistat que se us demana, haureu de fer un recorregut de la llista node per node.

Sortir del Casino

Aquesta opció implicarà demanar el DNI del jugador que vol abandonar el Casino i fer una eliminació d'un node de la llista ordenada. Podeu fer servir la funció `EliminarElement` de la que també teniu el pseudo-codi.

Tancar Casino

Aquesta opció equival a l'opció 'Sortir' de la versió 2.0 amb la diferència de que ara, a més d'alliberar la llista de jugadors, heu d'escriure al fitxer 'ComptesCasino.txt' l'estat de comptes del Casino al moment del tancament. Per tant, tot el que us vam indicar en l'enunciat de la pràctica del bloc 2 en referència al recorregut i l'alliberament de la llista dinàmica continua sent vàlid per aquesta versió.

4. Avaluació

Us recordem que, a l'igual que en la pràctica dels blocs 1 i 2, en l'avaluació de la pràctica és valorarà:

- El correcte funcionament del programa
- El correcte ús del llenguatge C
- Resposta a preguntes sobre el codi

Els dos primers punts són igual d'importants. Si una pràctica funciona correctament però no està ben implementada no s'aprovarà. D'igual forma, si una pràctica està ben implementada però no funciona bé, tampoc estarà aprovada.

MOLT IMPORTANT: La informació dels jugadors s'ha de guardar en una llista encadenada dinàmica ordenada per DNI. Aquest punt és imprescindible per aprovar la pràctica.

Les preguntes sobre el codi s'han de respondre correctament per demostrar que tots els components del grup han participat en la implementació de la pràctica entregada.

Altres punts que es tindran en compte positivament:

- Claredat i estructura del codi: tabulació del codi, separacions entre blocs del programa, inclusió de comentaris, noms dels identificadors de les variables...

- Presentació per pantalla: missatges clars, la pantalla s'esborra cada cop que es mostra el menú o s'entra a una opció, etc...

- Control d'errors en l'entrada de dades (p.ex. controlar que la data de naixement sigui vàlida)

- Millores sobre el que es demana per facilitar el funcionament del programa (Algunes idees: permetre més d'una aposta per jugador, demanar confirmacions abans de sortir del programa,…)

Abans de lliurar la pràctica comproveu que funciona correctament en tots els casos. Proveu totes les possibilitats per assegurar que funciona bé sempre.

Per provar la lectura dels fitxers inicials, el dia de l'avaluació se us proporcionaran dos fitxers diferents als que ara se us posen com exemple. Per tant, mentre implementeu la pràctica creeu-vos diferents fitxes de proves amb el format donat per comprovar que funciona sempre (per exemple, proveu que us llegeix bé els comptes del Casino encara que contingui valors negatius o proveu de carregar una llista de No Admesos més llarga de la que se us dóna com exemple).

5. Annex: Obtenció de la data i la hora del sistema

A continuació teniu la informació que proporciona el Help del Visual C++ sobre la funció 'localtime' que és la que necessitareu per obtenir la data i la hora del rellotge del sistema. Fixeu-vos en l'exemple on teniu remarcat en negreta el que realment us interessa pel que voleu que és obtenir la data i la hora.

localtime

Converts a time value and corrects for the local time zone.

struct tm *localtime(const time_t *_timer_);

Routine	Required Header	Compatibility
localtime	<time.h>	ANSI, Win 95, Win NT

For additional compatibility information, see Compatibility in the Introduction.

Libraries

LIBC.LIB	Single thread static library, retail version
LIBCMT.LIB	Multithread static library, retail version
MSVCRT.LIB	Import library for MSVCRT.DLL, retail version

Return Value

localtime returns a pointer to the structure result. If the value in *timer* represents a date before midnight, January 1, 1970, **localtime** returns **NULL**. The fields of the structure type tm store the following values, each of which is an **int**:

tm_sec Seconds after minute (0 – 59)

tm_min Minutes after hour (0 – 59)

tm_hour Hours after midnight (0 – 23)

tm_mday Day of month (1 – 31)

tm_mon Month (0 – 11; January = 0)

tm_year Year (current year minus 1900)

tm_wday Day of week (0 – 6; Sunday = 0)

tm_yday Day of year (0 – 365; January 1 = 0)

tm_isdst

Positive value if daylight saving time is in effect; 0 if daylight saving time is not in effect; negative value if status of daylight saving time is unknown. The C run-time library assumes the United States's rules for implementing the calculation of Daylight Saving Time (DST).

Parameter

timer

Pointer to stored time

Remarks

The **localtime** function converts a time stored as a time_t value and stores the result in a structure of type **tm**. The **long** value *timer* represents the seconds elapsed since midnight (00:00:00), January 1, 1970, coordinated universal time (UTC). This value is usually obtained from the **time** function.

gmtime, **mktime**, and **localtime** all use a single statically allocated **tm** structure for the conversion. Each call to one of these routines destroys the result of the previous call.

localtime corrects for the local time zone if the user first sets the global environment variable **TZ**. When **TZ** is set, three other environment variables (**_timezone**, **_daylight**, and **_tzname**) are automatically set as well. See _tzset for a description of these variables. **TZ** is a Microsoft extension and not part of the ANSI standard definition of **localtime**.

Note The target environment should try to determine whether daylight saving time is in effect.

Example

```
/* LOCALTIM.C: This program uses time to get the current time
 * and then uses localtime to convert this time to a structure
 * representing the local time. The program converts the result
 * from a 24-hour clock to a 12-hour clock and determines the
 * proper extension (AM or PM).
 */

#include <stdio.h>
#include <string.h>
#include <time.h>

void main( void )
{
        struct tm *newtime;
        char am_pm[] = "AM";
        time_t long_time;

        time( &long_time );                /* Get time as long integer. */
        newtime = localtime( &long_time ); /* Convert to local time. */

        if( newtime->tm_hour > 12 )        /* Set up extension. */
                strcpy( am_pm, "PM" );
        if( newtime->tm_hour > 12 )        /* Convert from 24-hour */
                newtime->tm_hour -= 12;    /*   to 12-hour clock.  */
        if( newtime->tm_hour == 0 )        /*Set hour to 12 if midnight. */
                newtime->tm_hour = 12;

        printf( "%.19s %s\n", asctime( newtime ), am_pm );
}
```

Output

```
Tue Mar 23 11:28:17 AM
```

Time Management Routines

See Also asctime, ctime, _ftime, gmtime, time, _tzset