Primera pràctica obligatòria

Programació 1 – Grau en Enginyeria Informàtica

Curs 2018-2019

Exercici 1 (5 punts)

Implementeu en el llenguatge de programació ANSI C++ un programa que, donada una seqüència de caràcters que representa un número en notació romana i acabada en *intro* ('\n'), determini quin és el seu equivalent en el sistema decimal. Els caràcters acceptats són els següents: 'I', 'V', 'X', 'L', 'C', 'D', 'M' i '\n'. Si rebem un caràcter que no correspon a la llista d'acceptats traurem un missatge d'error i acabarem el procés. El programa no cal que comprovi si el número en notació romana introduït és correcte, podem suposar que és correcte (segueix les regles de la numeració romana) i que és més petit o igual a 3.000.

Exemple 1 – Donada la seqüència "MMCDXLIV" el resultat del programa ha de ser el següent:

```
El numero es 2444.
```

Exercici 2 (5 punts)

Implementeu en el llenguatge de programació ANSI C++ un programa per jugar al joc de "imparell o parell". L'usuari començarà amb un capital inicial de 1.000 € i a cada tirada especificarà quina quantitat vol apostar. Seguidament, especificarà si vol apostar a imparell (1) o parell (2) introduint el número corresponent.

El programa ha de simular la jugada (tirar 2 daus de l'1 al 6) i mostrar el resultat per pantalla. Si guanya, és a dir, si la suma dels 2 daus és imparell o parell segons el que ha dit l'usuari, se li mostrarà la quantitat guanyada i el nou capital del que disposa després de sumar-li la quantitat apostada. En canvi, si perd, se li notificarà que ha perdut, la quantitat que ha perdut i se li mostrarà el capital del que disposa després de restar la quantitat que ha apostat.

Una vegada notificat el nou capital, es tornarà a demanar una quantitat per apostar i tornarà a jugar. Si posa l'opció de sortir (prem 0), el programa acabarà mostrant les vegades que ha guanyat, les vegades que ha perdut i el capital final.

L'usuari no pot apostar més del seu capital, i si ho perd tot, el programa acaba mostrant el missatge de que s'ha arruïnat juntament amb el número de partides guanyades i perdudes.

Es poden fer servir les llibreries time.h i stdlib.h per a la generació de números aleatoris tal com mostra l'exemple següent:

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main () {
     int n = 0;
     srand(time(NULL)); // Inicialització, només s'ha de fer una vegada al principi
     while (n != 4) { // En aquest cas acabarà quan surti el número 4
          n = rand() % 10 + 1; // Genera números de l'1 al 10
          printf("Numero generat %i\n", n);
     }
}
```

Exemple 1 – Donats els següents número entrats per l'usuari “100 2 200 1 700 1 0”, una possible sortida del teu programa pot ser (dependrà dels números que surtin aleatòriament):

```
Benvingut al joc de Imparell o Parell!
Tens un capital de 1000 Euros, quina quantitat vols apostar?
100
Introdueix el numero per la teva aposta (1 = imparell, 2 = parell):
2
Has apostat 100 Euros a 2 (parell)
Ha sortit 4 i 6 (suma = 10)
Has guanyat 100 Euros!
Tens un capital de 1100 Euros, quina quantitat vols apostar?
200
Introdueix el numero per la teva aposta (1 = imparell, 2 = parell):
1
Has apostat 200 Euros a 1 (imparell)
Ha sortit 6 i 6 (suma = 12)
Has perdut 200 Euros!
Tens un capital de 900 Euros, quina quantitat vols apostar?
700
Introdueix el numero per la teva aposta (1 = imparell, 2 = parell):
1
Has apostat 700 Euros a 1 (imparell)
Ha sortit 5 i 3 (suma = 8)
Has perdut 700 Euros!
Tens un capital de 200 Euros, quina quantitat vols apostar?
0
El teu capital final es de 200 Euros.
Has guanyat 1 partides i n'has perdut 2.
```

Exemple 2 – Donats els següents número entrats per l'usuari “3000 2000 500 1 200 5 -3 2 0”, una possible sortida del teu programa pot ser (dependrà dels números que surtin aleatòriament):

```
Benvingut al joc de Imparell o Parell!
Tens un capital de 1000 Euros, quina quantitat vols apostar?
3000
Tens un capital de 1000 Euros, quina quantitat vols apostar?
2000
Tens un capital de 1000 Euros, quina quantitat vols apostar?
500
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
1
Has apostat 500 Euros a 1 (imparell)
Ha sortit 3 i 6 (suma = 9)
Has guanyat 500 Euros!
Tens un capital de 1500 Euros, quina quantitat vols apostar?
200
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
5
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
-3
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
2
Has apostat 200 Euros a 2 (parell)
Ha sortit 1 i 2 (suma = 3)
Has perdut 200 Euros!
Tens un capital de 1300 Euros, quina quantitat vols apostar?
0
El teu capital final es de 1300 Euros.
Has guanyat 1 partides i n'has perdut 1.
```

Avaluació

- La pràctica s'ha d'implementar en el llenguatge ANSI C++ i s'ha d'executar correctament en una plataforma Linux. Per garantir que satisfà l'estàndard ANSI compileu afegint l'opció `-ansi`. És a dir, la comanda de compilació ha de ser:
  ```
  $ g++ programa.cpp -o executable -ansi
  ```
- La pràctica es puntua sobre 10 punts i el seu pes a l'avaluació final és del 15%.
- La data límit per al lliurament de la pràctica és el 18 de novembre.
- La pràctica es lliurarà via el Campus Virtual (CV), dins l'apartat Activitats.
- Es recomana posar comentaris dins els fitxers .cpp dels exercicis que ajudin a interpretar l'algorisme implementat.
- La pràctica s'ha de resoldre individualment o en grups de màxim 2 persones.
- Com a comentaris de l'activitat heu d'indicar si la pràctica s'ha realitzat de forma individual o en grup. A més heu d'indicar els membres que composen el grup. Finalment, comenteu breument l'estratègia emprada per resoldre cada problema.

- En cas de que la pràctica no es superi, podrà ser recuperada al realitzar la segona pràctica.

Aspectes a tenir en compte

Alguns aspectes que heu de tenir en compte a l'hora de realitzar les vostres pràctiques i que **puntuaran negativament** si no els teniu en compte, **encara que la pràctica *funcioni***:

- Nitidesa en el codi (tabulació correcta, no fer càlculs innecessaris, utilització correcta dels recursos de la màquina...).
- Utilitzar estructures algorísmiques adients per als problemes a resoldre.
- No es poden utilitzar taules per resoldre aquesta pràctica.
- Utilitzeu noms de variables entenedors.
- Llegiu atentament l'enunciat i no implementeu funcionalitats diferents a les que us demana.
- Queda prohibit utilitzar la instrucció de salt `goto`, o instruccions per alterar el funcionament normal d'un bucle com `continue` o `break` (l'únic cas en que es pot fer servir el `break` és en el cas que es faci anar la instrucció `switch`).
- No es poden utilitzar llibreries no estàndard com la `conio.h`.
- En cas de realitzar pràctiques de forma “col·laborativa” entre diferents grups, esmentar-ho en el moment de l'entrega o en els comentaris, encara que finalment s'entreguin les pràctiques per separat o individualment.
- Si es detecta que la pràctica és copiada, la nota és un 0, tant pel que còpia com pel que deixa copiar.

Mètode de correcció

Per a la validació del funcionament de la pràctica s'utilitzarà el Makefile i els corresponents jocs de proves. Tant els jocs de proves com el fitxer Makefile que s'utilitzarà per a la correcció els podreu trobar a la carpeta de la pràctica a l'apartat de Recursos del Campus Virtual. A l'hora de presentar la pràctica a través del CV, haureu d'enviar els següents fitxers:

- Makefile: El mateix fitxer “Makefile” que trobareu al CV.
- ex?-jp?.txt: Els fitxers de proves que trobareu al CV.
- ex?.cpp: Els fitxers on implementareu els vostres programes, un per cada exercici.

Per poder enviar tots els fitxers sense problemes els podeu comprimir utilitzant la comanda 'tar' de la següent forma:

Suposant que teniu els exercicis resolts dins la carpeta 'prac' podeu fer:
$ cd prac
$ tar cvfz prac.tgz *
El fitxer resultant és el fitxer prac.tgz que conté tots els fitxers de la carpeta 'prac'.

Comandes que se seguiran per la correcció:

1. $ make clean
   Per eliminar els possibles fitxers binaris que hi pugui haver.
2. $ make all
   Per compilar els exercicis seguint l'estàndard ANSI.
3. $ make test
   Per provar el correcte funcionament dels programes d'acord amb l'enunciat.

Exemple d'una sortida correcta després d'executar aquestes 3 comandes:

```
**** Exercici 1.0 ****
El numero es 2444.
```

```
**** Exercici 1.1 ****
El numero es 938.
**** Exercici 2.0 ****
Benvingut al joc de Imparell o Parell!
Tens un capital de 1000 Euros, quina quantitat vols apostar?
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
Has apostat 100 Euros a 2 (parell)
Ha sortit 6 i 1 (suma = 7)
Has perdut 100 Euros!
Tens un capital de 900 Euros, quina quantitat vols apostar?
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
Has apostat 200 Euros a 1 (imparell)
Ha sortit 4 i 2 (suma = 6)
Has perdut 200 Euros!
Tens un capital de 700 Euros, quina quantitat vols apostar?
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
Has apostat 700 Euros a 1 (imparell)
Ha sortit 1 i 5 (suma = 6)
Has perdut 700 Euros!
Ho has perdut tot!
Has guanyat 0 partides i n'has perdut 3.
**** Exercici 2.1 ****
Benvingut al joc de Imparell o Parell!
Tens un capital de 1000 Euros, quina quantitat vols apostar?
Tens un capital de 1000 Euros, quina quantitat vols apostar?
Tens un capital de 1000 Euros, quina quantitat vols apostar?
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
Has apostat 500 Euros a 1 (imparell)
Ha sortit 6 i 1 (suma = 7)
Has guanyat 500 Euros!
Tens un capital de 1500 Euros, quina quantitat vols apostar?
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
Introdueix el numero per la teva aposta (1 = imparell, 2 =
parell):
Has apostat 200 Euros a 2 (parell)
Ha sortit 4 i 2 (suma = 6)
Has guanyat 200 Euros!
Tens un capital de 1700 Euros, quina quantitat vols apostar?
El teu capital final es de 1700 Euros.
Has guanyat 2 partides i n'has perdut 0.
```