FUNDAMENTOS DE INFORMÁTICA

Examen Parcial de Teoría
Modelo B

1. Construid una función entero_sin_extremos(n) que devuelva el entero obtenido al eliminar la primera y la última cifra de n. Por ejemplo, entero_sin_extremos(82047)=204. Utilizad dicha función en un algoritmo que solicita un entero x al usuario, y escribe en la pantalla el número obtenido al eliminar la primera y la última cifra de x. [3 puntos]

2. Construid un algoritmo que encuentre el primer número que sea igual al producto de los anteriores en una secuencia de enteros negativos acabada en 0. El algoritmo debe utilizar una acción denominada lee_entero_negativo(n) que solicita al usuario que introduzca un número entero negativo y comprueba si el número introducido por el usuario es un entero negativo antes de asignarlo a n. Si el usuario introduce un número que no es un entero negativo, la acción lee_entero_negativo(n) vuelve a pedirle que introduzca un entero negativo tantas veces como sea necesario. [3.5 puntos]

3. Escriure un algorisme que llegeixi una frasse acabada en el caràcter '.' i escrigui si a la frasse hi ha més paraules que comencen en 'ca' (el caràcter 'c' seguit de 'a') que paraules que comencen en 'sa'. No és obligatori utilitzar subprogrames. [3.5 punts]

FONAMENTS D'INFORMATICA

Exàmen Parcial de Pràctica
Model B

1. Escriure un programa que llegeixi una seqüència d'enters positius acabada en 0 i digui quants imparells hi ha a la seqüència i si hi ha més numeros imparells acabats en 3 que números imparells acabats en 9. S'ha d'utilitzar i definir la funció que digui si un enter passat com a paràmetre d'entrada és parell o no ho és. [4 punts]

2. Objectiu: Farem un programa que donat n comprovará la seguent fórmula, calculant-ne els dos costats i veient que són iguals:

$$\frac{n!}{(n-0)!\cdot 0!} + \frac{n!}{(n-1)!\cdot 1!} + \frac{n!}{(n-2)!\cdot 2!} + ... + \frac{n!}{(n-m)!\cdot m!} + ... + \frac{n!}{(n-n)!\cdot n!} = 2^n$$

Detalls: Cal fer el programa complet. Necessitarem les següents peces:

a. `double Factorial(int n)`, que multiplica els nombres de 1 a n, convertits en *doubles*. Recorda que, per definició, $0! = 1$.

b. `double pow(double x,double y)`, que calcula x^y, i es pot usar si s'inclou math.h. Recorda passar n a *double* abans d'usar-la.

c. `main()`,

- que defineix un s *double*, i un c *double* auxiliar,
- demana a l'usuari un n *int*,
- i va sumant a s cada cop el terme

$$c = Factorial(n)/(Factorial(n-m) * Factorial(m))$$

- desde $m = 0$ fins $m = n$, i al acabar mostra s,
- despres calcula $2{,}0^n$, i també mostra el resultat.

3. Implementeu una funció en que li passem una quantitat en segons, i ens torna la quantitat de minuts correspondent. Recordeu que els minuts no tenen part decimal.

Creeu programa principal on demaneu a l'usuari la quantitat en segons a transformar, realitzeu la crida a la funció anterior passant-li aquesta quantitat, i mostreu finalment a l'usuari la quantitat de minuts. [2 punts]