

G4011105 – Programación I

1. Escribir un programa que declare e inicie (con calquera valor que queirades) tres variables: unha de tipo **float**, outra de tipo **int**, e outra de tipo **char**. O voso programa entón terá que imprimir o enderezo asociado a cada variable e o valor almacenado en cada unha deles. Utilizar o especificador de formato **"%p"** para imprimir os enderezos en notación hexadecimal (veredes enderezos do tipo: **0xbfe55918**, onde os caracteres iniciais **0x** indican notación hexadecimal e o resto dos díxitos dan o enderezo en base 16). É posible distinguir o tipo de dato a partires do valor do enderezo impreso? Se lle sumamos 1 a eses enderezos de memoria, que resultados obtemos e por que?

Os punteiros son variables que almacenan enderezos de memoria. Create agora dous punteiro de tipo **int*** e **float***. Asignádelle ao punteiro **int*** o enderezo de memoria da variable **int**, e ao **float*** o enderezo da variable de tipo **float**, e imprímelle con **printf** o valores accedidos mediante eses punteiros. Repetide agora a operación pero agora asignando a **int*** o enderezo da variable tipo **float**, a **float*** o enderezo da variable tipo **int**. Por que se obteñen valores distintos aos anteriores?

2. Repetide o último exercicio do boletín anterior, pero agora considerando estes novos prototipos de funcións:

WeatherData *Consulta_por data(File* file, Date *d) /*A esta función pásaselle unha data e devolve unha estrutura cos datos recollidos dese día*/

float Consulta_precip_media(File* file, Date *d1, Date *d2) /*A esta función pásaselle dúas datas e devolve a precipitación media*/

3. Consideremos un arquivo tipo **weatherdata-420-72.dat** (creado no boletín anterior) que contén información meteorolóxica en formato binario. Escribide un programa cunha función que lea todos os datos dese arquivo binario, entre dúas datas concretas pasadas como argumentos, os garde nun vector de estruturas **WeatherData** e devolva o punteiro a ese vector de estruturas. Facede reserva dinámica de memoria para ese vector (tamaño axustado ao número de entradas entre as datas indicadas). Imprimide os datos da primeira e última entrada dende a función **main()**, e finalmente liberade a memoria. Tede en conta que ademais dos datos é necesario saber cantos elementos ten o vector de estruturas.
4. Create un programa que lea un nome de arquivo e o almacene nunha estrutura con (só) dous campos: un para o nome base, e outro para a extensión. **O nome do arquivo debe pasarse por liña de comandos**, e a estrutura debe pasárselle como argumento a unha función que abra ese arquivo e imprima o seu contido por pantalla. A función ten que manexar os casos de nomes con extensión e sen ela.