

G4011105 – Programación I

1. A continuación preséntase unha función para o cálculo do promedio de dous números, xunto cun exemplo de invocación dende un programa principal:

```
float encontprom(int num1, int num2)
{
    float promedio;

    promedio = (num1 + num2) / 2.0;
    return(promedio);
}

main()
{
    int a,b;
    float resultado;
    printf("Dame dous valores enteiros: ");
    scanf("%d %d",&a,&b);

    resultado = encontprom(a, b);
    printf("Promedio=%f\n",resultado);
}
```

Engadir outra función para o cálculo do máximo de dous valores pasados como argumentos, e comprobar o seu funcionamento e imprimir o resultado dende a función `main()`.

2. Escribir unha función que imprima ao revés unha palabra pasada como argumento.
3. Escribir un programa cunha función que teña como parámetro formal unha cadea de caracteres, que substitúa todas as maiúsculas por minúsculas, e que devolva (usando o **return**) o número de cambios realizados. Non empreguedes funcións das bibliotecas estándares agás para lectura e escritura e tede en conta que a cadea de entrada pode conter letras maiúsculas e minúsculas, números e outros caracteres imprimibles (non consideredes acentos, nin diéreses). Imprimir dende o `main()` a cadea transformada, así coma o número de cambios realizados.
4. Escribir un programa que permitan realizar as seguintes operacións sobre polinomios de ata grao 5. Debe definirse unha estrutura (`t_polinomio`) para almacenar grao e coeficientes dun polinomio e as funcións cuxo prototipo se indica a continuación:

```
void Inicia (int grao, int coe[], t_polinomio *pol); //Inicia unha estrutura a partir dos datos
                                                    // do grao e dun vector de coeficientes

double Avaliar ( t_polinomio pol, double pt); //Devolve o valor do polinomo nun punto.

void Mostrar (t_polinomio pol); //Imprime os coeficientes non nulos do polinomio
void Derivar(t_polinomio pol, t_polinomio *der); //Calcula o polinomio derivado
```

Non se debe facer uso de variables globais e os resultados deben imprimirse invocando as funcións correspondentes dende a función `main()`. Lembrar que unha función non debe ler datos nin escribilos a non ser que se trate precisamente dunha función para lectura ou escritura de datos.

5. Escribir un programa cunha función que determine a media dos valores dunha secuencia de enteiros almacenada nun arquivo. Lede os datos do arquivo e ideos gardando nun vector. Facede a lectura ata chegar ao final o arquivo (nese momento **feof(ptf)!=0**). No arquivo cabeceira **stdlib.h** está definida a función de formato **int feof(FILE *ptf)**, que ten coma argumento o punteiro a arquivo e devolve un valor distinto de cero cando chega ao final do arquivo e cero noutro caso.
6. Escribir un programa que lea un arquivo de texto e que garde noutro arquivo ese texto encriptado da seguinte forma: o primeiro carácter imprimeo tal como foi lido, nos demais casos, cada carácter é

substituído polo carácter na posición da táboa ASCII que corresponde ao módulo **128** da suma do código ASCII do carácter actual máis o do carácter lido previamente. Exemplo: Se tivésemos no arquivo orixe:

35 Abacates

entón debe aparecer no arquivo de saída:

3hUa#CDDUYX

Empregar unha función para o encriptado.

7. Escribir un programa que lea un texto almacenado nun arquivo, encriptado de acordo ca técnica descrita antes, descifre tal texto e o presente descifrado por pantalla. Empregar unha función para o desencriptado. Para desencriptar hai que sumarlle 128 á resta de códigos e despois cacular o módulo 128.
8. Dados dous vectores U e V, defínese a distancia de Hamming, $d(U,V)$, coma o número de posicións nas que difiren os vectores. Por exemplo, se $U = 20001$ e $V = 10103$, $d(U,V) = 3$, xa que existen tres posicións nas cales os vectores teñen diferente valor. Escribide unha función, denominada Hamming, que reciba como parámetros dúas cadeas de caracteres e devolva un enteiro que indique a distancia Hamming entre elas. Se as cadeas non teñen lonxitudes iguais a función devolverá -1.

Empregar esa función dentro dun programa en C que pida ao usuario unha chave de paso (cadea de caracteres), e que a compare coas almacenadas nun arquivo denominado **chaves.dat** xa existente e que, despois de chamar á función Hamming, amose por pantalla unha mensaxe indicando se a cadea dada polo usuario coincide ou non con algunha das almacenadas no arquivo chaves.dat. Considerar que as chaves de paso do arquivo están encriptadas segundo o método anterior e que por tanto fai falta desencriptalas para facer a comparación.

9. Nun almacén precísase gardar os datos necesarios para os distintos artigos dispoñibles nun ficheiro chamado **invent.dat**. Para cada elemento deben rexistrarse os seguintes datos:

codigo (enteiro)
nome (cadea de ata 30 caracteres)
prezo (número real),
cantidade (enteiro que indica a cantidade de unidades existente).

Escribir unha función chamada **gravar_datos** que poida gardar os datos de varios artigos no arquivo **invent.dat**. A escritura debe facerse en binario, gardando unha estrutura en cada operación. O código para cada artigo será xerado secuencialmente a partir do 1 e os restantes datos serán tecleados polo usuario, mentres que desexa continuar coa gravación de máis artigos. Ao final, a función devolve o número de elementos inseridos.

Escribir unha función chamada **ler_datos** que cargue todos os datos do arquivo de inventario nun vector de estruturas, que se lle pase como argumento.

Escribir un procedemento, co nome **venta_item** que reciba tres parámetros: o código dun elemento, o número de unidades a seren vendidas e o vector de estruturas co inventario. Este procedemento debe acceder no vector de estruturas ao artigo que se quere vender e amosar na pantalla o valor total da venda. Se a cantidade de artigos existentes é menor ca cantidade esixida para a venda, debe mostrar unha mensaxe indicando que non se pode realizar a venda; noutro caso debe actualizar o número de unidades do artigo correspondente.

Escribir un programa en C para probar o funcionamento das tres funcións. Antes de rematar o programa debe actualizarse o arquivo **invent.dat**.