

Ejercicios del Tema 3

Sentencias Repetitivas

1. Mitades sucesivas

Realiza un programa que pida un número real, y presente por pantalla ese número y sus mitades sucesivas hasta que el valor sea menor que 1.

```
#include <stdio.h>

int main()
{
    float num;

    printf("Escribe un numero real.");
    scanf("%f", &num);

    while (num >= 1)
    {
        printf("%f\n", num);
        num = num / 2;
    }
}
```

2. Números comprendidos entre A y B

Realiza un programa que muestre por pantalla los números enteros comprendidos entre 1 y 100 contando de n en n. El valor de n será introducido por el usuario desde el teclado. Si el usuario introduce un valor menor que 1 el programa se lo volverá a pedir de forma reiterada hasta que el usuario introduzca un número mayor o igual que 1.

```
#include <stdio.h>

int main()
{
    int i, a = 0, b = 100, n;

    // Con un bucle do-while nos aseguramos de que el número n es mayor
    // que 1
    do
    {
        printf("Elige el valor de n: ");
        scanf("%d", &n);
    }
    while (n < 1);

    // Una vez que tenemos n simplemente imprimimos la secuencia
    // actualizando el indice del bucle for con el valor de n
    for (i = a; i <= b; i += n)
        printf("%d\t", i);
}
```

```

return 0;
}

```

3. Aproximación a π

Realiza un programa que calcule una aproximación al número π mediante el sumatorio de la serie:

$$\pi \simeq 4 \cdot \sum_{n=0}^{10^6} \frac{(-1)^n}{2n+1}$$

```

#include <stdio.h>
#include <math.h>

int main()
{
    double pi = 0;
    int n;

    for (n = 0; n <= 1e6; ++n)
    {
        pi += pow(-1, n) / (2 * n + 1);
    }
    // La multiplicacion se realiza fuera del bucle
    // para evitar repetir esa operacion 1 millon de veces
    pi = 4 * pi;

    printf("%lf", pi);

    return 0;
}

```

4. Sucesión de Fibonacci

Escribe un programa que genere los n primeros términos de la sucesión de Fibonacci. El número entero n deberá ser leído por teclado. Este valor debe ser positivo, de forma que si el usuario introduce un valor negativo el programa volverá a pedir que lo introduzca.

En la sucesión de Fibonacci los dos primeros números son 1, y el resto se obtiene sumando los dos anteriores: 1, 1, 2, 3, 5, 8, 13, 21, ...

```

#include <stdio.h>

int main()
{
    int i, n;
    int fib1 = 1, fib2 = 1;
    // La serie crece rapidamente, y un int no es suficiente
    unsigned long long int fib;

    // Con un bucle do-while nos aseguramos de que el número n es mayor
    // que 1
    do
    {
        printf("Elige el valor de n: ");
        scanf("%d", &n);
    }
    while (n < 1);
    // Aqui empieza el calculo de la serie
    for (i = 1 ; i <= n ; i++)

```

```

{
    // Los dos primeros terminos son 1
    if (i <= 2)
        fib = 1;
    else
    { // el resto de terminos son la suma de los dos anteriores
        fib = fib1 + fib2;
        // Una vez hecha la suma intercambio los valores para el
        // siguiente paso de bucle.
        fib1 = fib2;
        fib2 = fib;
    }
    // El identificador de formato esta adaptado a la clase del dato
    printf("%llu\t", fib);
}
return 0;
}

```

5. Números primos

Realiza un programa que muestre por pantalla los números primos comprendidos entre el 1 y el 300. Recuerda que un número entero es primo si es divisible únicamente por sí mismo y por la unidad.

```

#include <stdio.h>

int main()
{
    int n, k;
    // variable que controla si es primo o no
    _Bool esPrimo;

    for(n = 2; n <= 300; n++)
    {
        // Suponemos que el numero es primo
        esPrimo = 1;
        for(k = 2; k < n; k++)
        {
            // Si es divisible por algun numero deja de ser primo
            if (n % k == 0)
                esPrimo = 0;
        }
        // Si finalmente es primo, lo muestra en pantalla
        if (esPrimo)
            printf("%d\n", n);
    }
    return 0;
}

```

6. Velocidad de un objeto

Escribe un programa que calcule la distancia que recorre un objeto y la velocidad alcanzada cada 0.5 segundos durante los 10 primeros segundos de su movimiento. El objeto se mueve desde la posición cero, con velocidad inicial nula y con una aceleración constante tecleada por el usuario.

```

#include <stdio.h>

int main()
{
    int n;

```

```

float d, v = 0, a;
// Resolucion temporal
float dt = 0.5;

printf("Indica aceleracion.\n");
scanf("%f", &a);

for (n = 1; n <= 20; ++n)
{
    // Calculamos la distancia en cada intervalo, por tanto no se
    // incrementa
    d = v * dt + 0.5 * a * dt * dt;
    // La velocidad va incrementando
    v += a * dt;
    printf("d = %.2f, v = %.2f (t = %.2f)\n",
        d, v, n * dt);
}
}

```

7. Números aleatorios

Escribe un programa que sirva para jugar a adivinar un número. El ordenador genera un número aleatorio comprendido entre 1 y 100. El usuario debe adivinarlo y tras cada intento recibe una indicación si el número introducido es mayor, menor o igual al número a adivinar.

Para generar el número aleatorio se emplea el siguiente código:

```

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

int main()
{
    int n;
    ...
    //Genera semilla aleatoria
    srand(time(NULL));
    //Numero aleatorio entre 1 y 100
    n = rand() % 100 + 1;
    ...
}

```

```

#include <stdio.h>
//Para srand y rand
#include <stdlib.h>
//Para time
#include <time.h>

int main()
{
    int n, num, intentos = 0;

    //Genera semilla aleatoria
    srand(time(NULL));
    //Numero aleatorio entre 1 y 100
    n = rand() % 100 + 1;
    printf("Estoy listo. Di un numero:\n");
    do
    {

```

```
scanf("%d", &num);

if(num > n)
    printf("Muy alto\n");
else if (num < n)
    printf("Muy bajo\n");

++intentos;
}
while (num != n);

printf("Bien, era el %d\n", num);
printf("Has necesitado %d intentos.\n", intentos);
}
```