

PRÁCTICAS PARA LA ASIGNATURA

FUNDAMENTOS DE INFORMÁTICA

PRÁCTICA 5:
Estructuras de repetición.

INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Pedro María Alcover Garau
Dolores Cano Gil
Carlos Fernández Andrés
Pedro Javier Navarro Lorente



PRÁCTICA QUINTA

1. Visualizar los 100 primeros números naturales. Solución con tres bucles diferentes.

Solución n. 1:

```
#include<stdio.h>
int main(void)
{
    short int iNum;
    printf("los 100 primeros naturales son:\n");
    iNum = 1;
    do
    {
        printf("%hd", iNum);
        iNum++;
    }while(iNum <= 100);
    return 0;
}
```

Solución n. 2:

```
#include<stdio.h>
int main(void)
{
    short int iNum;
    printf("los 100 primeros naturales son:\n");
    iNum = 1;
    while(iNum <= 100)
    {
        printf("%hd", iNum);
        iNum++;
    }
    return 0;
}
```

Solución n. 3:

```
#include<stdio.h>
int main(void)
{
    short int n;
    printf("los 100 primeros naturales son:\n");
    for (n = 1 ; n <= 100 ; n++) printf("%hd", n);

    return 0;
}
```



Práctica nº 5: Estructuras de repetición. **3 de 12**

- 2. Calcula y visualiza una potencia. Para ello solicita la base y el exponente.**

```
#include<stdio.h>

int main(void)
{
    short int iBase, iExp, x;
    long int iRes = 1;

    printf("*** CALCULO DE UNA POTENCIA ***\n");
    printf("\nTeclea la Base: ");
    scanf("%hd", &iBase);
    printf("\nTeclea el Exponente: ");
    scanf("%hd", &iExp);

    /* Bucle "para" para el cálculo */
    for(x = 1 ; x <= iExp ; x++) iRes *= iBase;

    printf("El Resultado es: %ld", iRes);
    return 0;
}
```

- 3. Visualizar la tabla de multiplicar de un número introducido por teclado.**

```
#include <stdio.h>

int main(void)
{
    short int N, c;

    printf("Indicar qué tabla se quiere mostrar: ");
    scanf("%hd", &N);

    for(c = 1 ; c <= 10 ; c++)
        printf("%hd * %hd = %hd\n", N, c, N * c);

    return 0;
}
```

- 4. Hacer un programa que calcule la suma de los 10 primeros números pares y el producto de los 10 primeros números impares simultáneamente. Ejemplo de utilizar "switches" para par e impar.**



Práctica nº 5: Estructuras de repetición. 4 de 12

```
#include <stdio.h>

int main(void)
{
    short int iSP, iPi, c, Sw;

    // Inicializamos iSP al elemento neutro de la suma.
    iSP = 0;
    // Inicializamos iPi al elemento neutro del producto.
    iPi = 1;
    c = 0;
    Sw = 0;    // Cambiará de 0 a 1 alternativamente.

    do
    {
        c++;
        if(!Sw)
        {
            iPi *= c;
            Sw = 1;
        }
        else
        {
            iSP += c;
            Sw = 0;
        }
    }while(c<20);    // contamos desde 0 hasta 19.

    printf("Suma = %hd\tProducto = %hd", iSP, iPi);
    return 0;
}
```

5. Calcular y visualizar el factorial de números enteros introducidos por teclado, el programa terminará cuando se introduzca un 0.

```
#include <stdio.h>

int main(void)
{
    short int N;;
    long int F;

    /* este programa funciona mientras se introduzca un
    número distinto de cero visualizando su factorial. */

    printf("\n di un numero: ");
    scanf("hd", &N);

    while (N != 0)
    {
```



Práctica nº 5: Estructuras de repetición. 5 de 12

```
F = 1;
    for (i = 1 ; i <= N ; i++) F *= i;
/* Fuera del bucle for visualiza su factorial*/
/* El rango de un entero largo es desde -2147483648
hasta +2147483647. El factorial de 14 ya supera
ese rango de valores. ----- */
    printf("\n el factorial de %hd es %ld", N, F);

    printf("\n di un numero: ");
    scanf("hd", &N);
}

return 0;
}
```

Otro modo de hacerlo (quizá más elegante, aunque menos intuitivo):

```
#include <stdio.h>

int main(void)
{
    short int N;
    long int F;

    do
    {
        printf("\n di un numero: ");
        scanf("hd", &N);
        if(!N) break;

        for(F = 1, i = 1 ; i <= N ; F *= i++);

        printf("\n el factorial de %hd es %ld", N, F);
    }while(1);

    return 0;
}
```

6. Calcular la media de una serie de números enteros introducidos por teclado. El programa termina cuando en número introducido sea cero.

```
#include <stdio.h>

int main(void)
{
    short int N;           // Sucesivos números de entrada.
    short int cont = 0;    // Cantidad números introducidos.
    long int Suma = 0;     // Suma acumulada de los números.

    /* El programa funciona mientras se introduzca un numero
```



Práctica nº 5: Estructuras de repetición. 6 de 12

distinto de cero. Visualiza la media de los números
introducidos ----- */

```
do
{
    printf("Introduce número (%hd): ", cont + 1);
    scanf("%hd", &N);
    if(!N) break;
    iSuma+=iNum;
    cont++;
}while(1);
/* fuera del bucle calcula y visualiza */
printf("\nSe han introducido %hd números", cont);
if(cont) printf("\nMedia: %8.2lf", (double)Suma/cont);

return 0;
}
```

7. Calcular y visualizar el resultado de la siguiente serie, siendo a y b dos números enteros sin signo introducidos por teclado:

$$f(a,b) = \frac{a^1 + b^2 + a^3 + b^4 + \dots + a^{99} + b^{100}}{1 + 2 + 3 + 4 + \dots + 99 + 100}$$

Solución con DOS bucles:

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    short int a,b, den;
    double num,res;

    printf("Introduzca el valor de a: ");
    scanf("%hd", &a);
    printf("Introduzca el valor de b: ");
    scanf("%hd", &b);

    num = den = res = 0;
    for (i = 1 ; i <= 99 ; i += 2)
        num += (pow(a , i)) + (pow(b , i + 1));

    for (i = 1 ; i <= 100 ; i++)
        den += i;

    /* fuera del bucle calcula y visualiza */
    res = num / den;
    printf("\n f(%hd, %hd) = %lf", a, b, res);

    return 0;
}
```



Práctica nº 5: Estructuras de repetición. **7 de 12**

Solución con UN SOLO bucle:

En este caso, las dos estructuras for quedarían en una sola:

```
for (i = 1 ; i <= 99 ; i += 2)
{
    num += (pow(a , i)) + (pow(b , i + 1));
    den += 2 * i + 1;
}
```

Y el resto del código es igual que en el caso anterior.

8. Visualizar si un número entero introducido por teclado es perfecto. Terminar cuando en número introducido sea cero.

Un número se dice que es perfecto cuando es igual a la suma de todos sus divisores (incluido el 1) excluido él mismo. Por ejemplo, $28 = 1 + 2 + 4 + 7 + 14$, luego 28 es un número perfecto.

```
#include <stdio.h>

int main(void)
{
    long int N, Suma, div;
    /* Este programa funciona mientras se introduzca un
    numero distinto de cero. Indica si es perfecto o no.*/
    do
    {
        printf("Introduzca un número a testear... ");
        scanf("%ld", &N);
        if(!N) break;
        for(Suma = 1, div = 2 ; i <= N/2 ; i++)
            if (N % div == 0) Suma += div;
        if (N == Suma) printf("%ld es PERFECTO", N);
        else printf("%ld NO es PERFECTO", N);
    }while(1);

    return 0;
}
```

9. Visualizar todos los números perfectos comprendidos entre a y b (números enteros introducidos por teclado, debiendo ser a menor que b y mayor que 2).



Práctica nº 5: Estructuras de repetición. 8 de 12

```
#include <stdio.h>

int main(void)
{
    short int N,div,Suma,a,b;
    printf("Búsqueda de perfectos entre a y b\n\n");
    do
    {
        printf("\nValor de a (mayor o igual que 2): ");
        scanf("%hd", &a);
    }while(a < 2);
    do
    {
        printf("\nValor de b (mayor que a): ");
        scanf("%hd", &b);
    }while(b <= a);

    printf("Los perfectos entre %hd y %hd son:\n", a, b);
    for (N = a ; N <= b ; N++) // N: todos los candidatos.
    {
        for (Suma = 1 , div = 2 ; div <= N / 2 ; div++)
            if (N % div == 0) Suma += div;
        /* fuera del bucle for segundo*/
        if (N == Suma) printf("%10hd", N);
    }

    return 0;
}
```

10. Introducir un número entero por teclado y visualizar si es primo. Terminar cuando en número introducido sea cero.

```
#include <stdio.h>
int main(void)
{
    short int N, div;

    do
    {
        printf("Introduzca un entero ... ");
        scanf("%hd", &N);
        if(!N) break;
        for(div = 2 ; div <= sqrt(N) ; div++)
            if(N % div == 0) break;
        if(N%div == 0) printf("%hd es COMPUESTO.\n", N);
        else printf("%hd es PRIMO.\n", N);
    }while(1);

    return 0;
}
```




Práctica nº 5: Estructuras de repetición. **9 de 12**

Obsérvese que la sentencia condicional que se encuentra en el único bucle **for** de la solución podría haberse introducido dentro del propio **for**, quedando entonces el código de la siguiente forma:

```
#include <stdio.h>
int main(void)
{
    short int N, div;
    /* Este programa funciona mientras se introduzca un
    número distinto de cero ----- */
    do
    {
        printf("Introduzca un entero ... ");
        scanf("%hd", &N);
        if(!N) break;
        for(div = 2 ; div <= sqrt(N) && N % div ; div++);
        if(N % div) printf("%hd es PRIMO.\n", N);
        else printf("%hd es COMPUESTO.\n", N);
    }while(1);

    return 0;
}
```

Obsérvese también que en ese caso la estructura iterativa se ha quedado si sentencia por iterar. Por eso hemos tenido que añadir una sentencia vacía (un punto y coma después de la estructura **for**).

11. Visualizar todos los números primos comprendidos entre a y b (enteros introducidos por teclado). El número a debe ser mayor que 2 y menor que b.

```
#include <stdio.h>

int main(void)
{
    short int N,div,a,b;
    /* El programa visualiza los primos entre a y b.      */
    printf("Búsqueda de primos entre a y b\n\n");
    do
    {
        printf("\nValor de a (mayor o igual que 2): ");
        scanf("%hd", &a);
    }while(a < 2);
    do
    {
        printf("\nValor de b (mayor que a): ");
```



Práctica nº 5: Estructuras de repetición. **10 de 12**

```
        scanf("%hd", &b);
    }while(b <= a);

    printf("Los primos entre %hd y %hd son:\n", a, b);

    for(N = a ; N <= b ; N++) // N: todos los candidatos.
    {
        for(div = 2 ; div <= sqrt(N) && N % div ; div++);
        if(N % div == 0) printf("%10hd", N);
    }

    return 0;
}
```

- 12. Escribir un programa que lea números y escriba cuántos números positivos y negativos se han leído. La finalización de la entrada de datos se realiza por la respuesta al siguiente mensaje "Terminar S/N?", escrito después de introducir cada número.**

```
#include <stdio.h>

void main(void)
{
    short int p, n, N;
    char opcion;

    p = 0;           // Positivos introducidos.
    n = 0;           // Negativos o Ceros introducidos.

    do
    {
        printf("Introducir Entero ... ");
        scanf("%hd", &N);
        if(N > 0) p++;
        else n++;

        printf("Terminar (S / N)?: ");
        do
        {
            // flushall();
            opcion = getchar();
            while(opcion != 'N' && opcion != 'S');
        }while(opcion == 'S');
        printf("Positivos: %hd\tNegativos: %hd", p, n);
    }
}
```



Práctica nº 5: Estructuras de repetición. **11 de 12**

13. Hacer un programa que lea N (N mayor o igual que 2) números enteros (tanto positivos como negativos) y muestre el mayor y el menor de los números leídos.

```
#include <stdio.h>

int main(void)
{
    short int j, N, Mayor, Menor, Numero;

    j = 1;
    printf("Valor ... ");
    scanf("%hd", &Mayor);
    Menor = Mayor;

    do
    {
        printf("Valor ... ");
        scanf("%hd", &Numero);
        j++;
        if(Numero > Mayor) Mayor = Numero;
        else if(Numero < Menor) Menor = Numero;
    }while(j < Num);

    printf("Mayor = %hd\tMenor = %hd", Mayor, Menor);

    return 0;
}
```

14. Visualizar la tabla de multiplicar de 9.

15. Visualizar la suma de los 200 primeros números pares.

16. Visualizar el abecedario en minúsculas y su equivalente numérico ASCII.

17. Determinar las situaciones de overflow en el ejercicio resuelto 5.6, realizando las pruebas con diferentes tipos de datos para la variable fac.

18. Calcular y visualizar el resultado de la siguiente serie, siendo a un numero entero sin signo introducido por teclado:

$$f(a) = \frac{a^0}{2} + \frac{a^1}{4} + \frac{a^2}{6} + \dots + \frac{a^{100}}{202}$$



Práctica nº 5:
Estructuras de repetición. **12 de 12**

19. Introducir una serie de 30 letras por teclado. Visualizar cuantas 'a', 'e', 'i', 'o', 'u' (mayúscula y minúsculas) tiene.

20. Realizar un programa para que enseñe una tabla de multiplicar: Solicitará que tabla se desea aprender. Solicitará que introduzca la respuesta a cada elemento de la tabla. Dará mensajes de correcto e incorrecto (en este caso solicitará que lo vuelva a repetir dando tres oportunidades) . El programa terminará cuando la tabla solicitada sea cero.

21. Calcula y visualiza los divisores de un número entero introducido por teclado y el número de divisores que tiene.

22. Calcular el número e mediante el siguiente desarrollo en serie: N es entero introducido por teclado.

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{N!}$$

con una aproximación A ($A > 1 / N!$). Por ejemplo, dado $A = 0.000001$, escribir $e = 2.71828198$ con aproximación de 0.000001

23. Escribir una tabla con los N primeros números triangulares. Los números triangulares se obtienen sumando los números naturales sucesivos 1, 2, 3, 4, ... ($1=1$, $3=1+2$, $6=1+2+3$, $10=1+2+3+4$, ...).

Por ejemplo, dado $N = 5$, escribir:

Tabla de 5 números triangulares

1
3
6
10
15

24. Sumar los N primeros términos de la siguiente sucesión: $1 + 2 \cdot x + 3 \cdot x^2 + 4 \cdot x^3 + \dots + N \cdot x^{(N-1)}$. Por ejemplo, dado $X = 2$ y $N = 3$, escribir: La suma de los 3 primeros términos es 17

25. Calcular el número de combinaciones de N elementos tomados de M en M, según la siguiente fórmula: $(N \text{ sobre } M) = N! / M! \cdot (N - M)!$

Por ejemplo, dado $N = 5$ y $M = 3$, escribir "Hay 10 combinaciones de 5 elementos tomados de 3 en 3".