

Programación I

Ejercicios – Bucles anidados

Iván Cantador

Escuela Politécnica Superior

Universidad Autónoma de Madrid

- Ejercicio 1: cronómetro de cuenta atrás
- Ejercicio 2: longitudes de las palabras de una frase
- Ejercicio 3: búsqueda de un patrón en una frase
- Ejercicio 4: exponencial de un número
- Ejercicio 5: ordenación de una lista de números

Ejercicio 1: cronómetro de cuenta atrás

2

Implementar en C un programa que pida al usuario un crono de la forma “mm:ss”, indicando unos minutos (que pueden ser 0) y segundos concretos, y que muestre por pantalla línea a línea el valor de ese crono disminuyendo segundo a segundo hasta llegar a 0:0.

```
#include <stdio.h>

int main() {
    int m, minuto, s, segundo;

    printf("Introduzca el tiempo inicial (mm:ss): ");
    scanf("%d:%d", &minuto, &segundo);

    for ( m=minuto; m>=0; m-- ) {
        for ( s=segundo; s>=0; s-- ) {
            printf("%02d:%02d\n", m, s);
        }
        segundo = 59;
    }
}
```

Ejercicio 2: longitudes de las palabras de una frase ⁴

Implementar en C un programa que pida al usuario una frase y escriba por pantalla las longitudes de las palabras que esa frase contiene.

```
#include <stdio.h>
#include <string.h>

#define MAX_CAD 128

int main() {
    char frase[MAX_CAD];
    int i, j, lFrase, lPalabra;

    printf("Introduzca una frase: ");
    gets(frase);

    lFrase = strlen(frase);

    for ( i=0; i<lFrase; i++ ) {
        lPalabra = 0;
        for (j=i; j<=lFrase; j++ ) {
            if( frase[j] == ' ' || frase[j] == '\t' || frase[j] == '\0' ) {
                printf("Palabra de longitud %d\n", lPalabra);
                i = j;
                break;
            }
            lPalabra++;
        }
    }

    return 0;
}
```

Ejercicio 3: búsqueda de un patrón en una frase

6

Implementar en C un programa que pida al usuario una frase y una palabra (patrón), y que muestre por pantalla las posiciones de la frase en la que la palabra aparece, si es que las hubiese.

```
#include <stdio.h>
#include <string.h>

#define MAX_CAD 128

int main() {
    char frase[MAX_CAD], patron[MAX_CAD];
    int i, j, encontrado;

    printf("Introduzca una frase: ");
    gets(frase);

    printf("Introduzca un patrón: ");
    gets(patron);

    for ( i=0; i<strlen(frase); i++ ) {
        encontrado = 1;
        for ( j=0; j<strlen(patron); j++ ) {
            if (frase[i+j] != patron[j] ) {
                encontrado = 0;
                break;
            }
        }
        if ( encontrado == 1 ) {
            printf("Patrón encontrado en la posición %d.\n", i);
        }
    }
}
```


Ejercicio 4: exponencial de un número

8

Implementar en C un programa que calcule el valor de la función exponencial en un punto x , es decir $\exp(x)$.

Para ello, el programa ha de aproximar $\exp(x)$ con (p.e., $n=20$ términos de) el desarrollo de Taylor:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}, \forall x; n \in \mathbb{N}_0$$

```

#include <stdio.h>

int main() {
    double x, exp, num, den;
    int i, n;

    printf("Introduzca el valor de x: ");
    scanf("%lf", &x);

    exp = 0;
    num = 1;
    den = 1;

    for (n=0; n<20; n++) {
        // Se calcula el numerador del término n-ésimo
        num = 1;
        for ( i=1; i<=n; i++ ) {
            num = num * x;
        }

        // Se calcula el denominador del término n-ésimo
        den = 1;
        for ( i=2; i<=n; i++ ) {
            den = den * i;
        }

        // Se acumula el término n-ésimo
        exp = exp + num / den;
    }

    printf("exp(%lf) = %lf\n", x, exp);

    return 0;
}

```

Ejercicio 5: ordenación de una lista de números

10

Implementar en C un programa que pida al usuario una lista de números y la ordene usando el “algoritmo de selección”.

50	26	7	9	15	27
----	----	---	---	----	----

 Array original

7	26	50	9	15	27	Se coloca el 7 en primera posición. Se intercambian 7 y 50
7	9	50	26	15	27	Se coloca el 9 en segunda posición. Se intercambian 9 y 26
7	9	15	26	50	27	Se coloca el 15 en tercera posición. Se intercambian 15 y 50
7	9	15	26	50	27	El menor de los que quedan es el 26. Se deja en su posición
7	9	15	26	27	50	Se coloca el 27 en quinta posición. Se intercambian 27 y 50

```

#include <stdio.h>

#define MAX_DIM 128

int main() {
    int lista[MAX_DIM], longitud, i, j, min, pmin, temp;

    printf("Introduzca la longitud de la lista de números a crear: ");
    scanf("%d", &longitud);

    for ( i=0; i<longitud; i++ ) {
        printf("Introduzca el %d-ésimo número de la lista: ", i);
        scanf("%d", &lista[i]);
    }

    for ( i=0; i<longitud; i++ ) {
        min = lista[i];
        pmin = i;

        // Se busca el mínimo (valor y posición) a partir de la posición i
        for ( j=i+1; j<longitud; j++ ) {
            if ( lista[j] < min ) {
                min = lista[j];
                pmin = j;
            }
        }

        // Se intercambian los números en las posiciones i, pmin
        temp = lista[i];
        lista[i] = lista[pmin];
        lista[pmin] = temp;
    }

    printf("Lista ordenada:\n");
    for ( i=0; i<longitud; i++ ) {
        printf("%d ", lista[i]);
    }
    printf("\n");
    return 0;
}

```