

**Programación avanzada: P2 – Conociendo C**

Diferencias entre C99 y C11

	C99	C11
Concurrencia y paralelismo	Ninguno	Tiene soporte con headers nuevos como <threads.h>
Estructuras y uniones anónimas	No hay	Permite estructuras y uniones anónimas dentro de estructuras declaradas
Aserción estática	Cuenta con #if y #error pero son preprocesadores, es decir, no tienen noción de tipos cuando se corren.	Cuenta con aserción estática formal y se ejecuta después que #if y #error, por lo que es más confiable.
Macros con expresiones genéricas	No hay	Permite agregar tipos genéricos a macros, para que cambien dependiendo del tipo de dato dado por parámetros.
Codificación UTF-8, UTF-16 y UTF-32	Solo tiene UTF-8	Tiene los tres. Introduce tipos nuevos de char para que sea más fácil definirlos: char16_t, char32_t

En resumen, C11 arregla varias cosas que faltaban en C99, y se considera un estándar más “safe” que su antecedente.

## Fibonacci

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

void printFibonacci(int n) {
    int* dp = malloc(sizeof(int) * (n + 1));
    dp[0] = 0;
    dp[1] = 1;

    for (int i = 2; i <= n; i++) {
        dp[i] = dp[i - 1] + dp[i - 2];
    }

    for (int i = 1; i <= n; i++) {
        printf("%i ", dp[i]);
    }
    printf("\n");

    free(dp);
}

int main(int argc, const char** argv) {
    int n = 0;
    printf("Enter how many fibonacci numbers you want: ");
    scanf("%i", &n);

    while (n <= 0) {
        printf("Please enter number greater than 0: ");
        scanf("%i", &n);
    }

    printFibonacci(n);
}
```

```
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter how many fibonacci numbers you want: 2
1 1
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter how many fibonacci numbers you want: 3
1 1 2
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter how many fibonacci numbers you want: 4
1 1 2 3
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter how many fibonacci numbers you want: 20
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
```

## Is Prime

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

bool isPrime(int num) {
    if (num <= 1) {
        return false;
    }

    for (int i = 2; i < num; i++) {
        if (num % i == 0) {
            return false;
        }
    }

    return true;
}

int main(int argc, const char** argv) {
    int num = 0;
    printf("Enter number to check if prime: ");
    scanf("%i", &num);

    printf("%s", isPrime(num) ? "True" : "False");
}
```

```
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter number to check if prime: 1
False
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter number to check if prime: 2
True
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter number to check if prime: 11
True
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter number to check if prime: 71
True
PS C:\Users\cbarr\Google Drive\ITC\Sexto Semestre\Programación avanzada\P2 - Conociendo C> ./a.exe
Enter number to check if prime: 70
False
```