

## Actividad 23/05/2023. Tres décimas

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

```
#define Tamano 10
```

```
int main ()
```

```
{
```

```
    char c;
```

```
    int x;
```

```
    int ARREGLO [Tamano];
```

```
    double y;
```

```
    printf ("El tamano en bytes de una variable char es: %lu\n", sizeof(c));
```

```
    printf ("El tamano en bits de una variable char es: %lu\n", (8*sizeof(c)));
```

```
    printf ("El tamano en nibble de una variable char es: %lu\n", (sizeof(c)*2));
```

```
    printf ("El tamano en bytes de una variable int es: %lu\n", sizeof(x));
```

```
    printf ("El tamano en bits de una variable int es: %lu\n", (8*sizeof(x)));
```

```
    printf ("El tamano en nibble de una variable int es: %lu\n", (sizeof(x)*2));
```

```
    printf ("\nEl tamano en bytes de un arreglo es: %lu\n", sizeof(ARREGLO[Tamano]));
```

```
    printf ("\nEl tamano en bits de un arreglo es: %lu\n", (8*sizeof(ARREGLO[Tamano])));
```

```
    printf ("\nEl tamano en nibble de un arreglo es: %lu\n", (sizeof(ARREGLO[Tamano])*2));
```

```
    printf ("El tamano en bytes de una variable double es: %lu\n", sizeof(y));
```

```
    printf ("El tamano en bits de una variable double es: %lu\n", (8*sizeof(y)));
```

```
    printf ("El tamano en nibble de una variable double es: %lu\n", (sizeof(y)*2));
```

```
    return 0;
```

```
}
```

## Convertir el siguiente código a Python

```
import sys
```

```
Tamano = 10
```

```

def main():
    c = "
    x = 0
    ARREGLO = [0] * Tamano
    y = 0.0

    print(f"El tamaño en bytes de una variable char es: {sys.getsizeof(c)}")
    print(f"El tamaño en bits de una variable char es: {sys.getsizeof(c) * 8}")
    print(f"El tamaño en nibble de una variable char es: {sys.getsizeof(c) * 2}")

    print(f"El tamaño en bytes de una variable int es: {sys.getsizeof(x)}")
    print(f"El tamaño en bits de una variable int es: {sys.getsizeof(x) * 8}")
    print(f"El tamaño en nibble de una variable int es: {sys.getsizeof(x) * 2}")

    print(f"\nEl tamaño en bytes de un arreglo es: {sys.getsizeof(ARREGLO)}")
    print(f"\nEl tamaño en bits de un arreglo es: {sys.getsizeof(ARREGLO) * 8}")
    print(f"\nEl tamaño en nibble de un arreglo es: {sys.getsizeof(ARREGLO) * 2}")

    print(f"El tamaño en bytes de una variable double es: {sys.getsizeof(y)}")
    print(f"El tamaño en bits de una variable double es: {sys.getsizeof(y) * 8}")
    print(f"El tamaño en nibble de una variable double es: {sys.getsizeof(y) * 2}")

```

Explicar porqué arroja números distintos con fuentes.

Los códigos traducidos de C a Python pueden arrojar números diferentes debido a varias razones:

Diferentes implementaciones de los lenguajes: C y Python son lenguajes diferentes con implementaciones distintas. Por lo tanto, las funciones y operaciones pueden comportarse de manera ligeramente diferente, lo que puede resultar en valores diferentes.

Diferencias en el manejo de tipos: C es un lenguaje de programación de bajo nivel que permite un control más preciso sobre los tipos de datos. En Python, los tipos de datos son más flexibles y dinámicos. Al realizar conversiones de tipos o cálculos, Python puede manejar automáticamente conversiones implícitas que pueden afectar los resultados numéricos.

Tamaño y representación de datos: En C, el tamaño de los tipos de datos puede variar según la plataforma y la implementación del compilador. Por otro lado, Python tiene un modelo de objeto más complejo y tiene un tamaño de objeto fijo más grande debido a la sobrecarga asociada con la representación de objetos. Esto puede afectar los resultados obtenidos al medir el tamaño de los tipos de datos.

Es importante tener en cuenta estas diferencias al traducir códigos de C a Python y no esperar resultados idénticos. Si necesitas resultados precisos y consistentes, es recomendable ajustar y adaptar el código según las características y particularidades de cada lenguaje.

Nolasco Valenzuela, J. S. (2018). *Python: aplicaciones prácticas..* RA-MA Editorial.  
<https://bibliodig.uag.mx:2139/es/lc/bibliouag/titulos/106523>