

# Tarea 01

Sinencio granados Dante Jusepee  
316246019

## 1) ¿Qué diferencias hay entre Java y C?

Aunque estos lenguajes comparten varias cosas en común como la sintaxis o el uso de bibliotecas, se puede ver que se diferencian en:

- Java: Es puramente orientado a objetos, el programa se ejecuta sobre una máquina virtual machine creada específicamente para java (JVM), el código debe ser interpretado después de ser compilado y se libera la memoria de forma "automática".
- C: Se pueden hacer programas no orientados a objetos o hasta combinado entre orientado y no orientado a objetos, se tiene control sobre el procesador, el programador se encarga de la liberación de memoria y se utilizan los punteros con una aritmética ya definida para la exploración de la memoria física.

---

## 2) ¿Que es el preprocesador y que salida genera?

El preprocesador es un programa que forma parte del compilador el cual modifica el código antes de ser "traducido" a código binario. El cambio que hace se basa en interpretar las líneas que empiezan con "#", esto quiere decir que acepta como entrada código fuente y da como salida otro código donde los macros ya fueron interpretados y agregados en las variables.

---

## 3) Para cada secuencia de instrucciones, indica si es correcto o no; en caso de que no lo sea, explica la razón.

a)

```
int main() {  
    long int a = 34 ;  
    char b = 'A' ;  
    long c = a * b ;  
    return 0 ;  
}
```

Para este caso la asignación de las variables se podrían ver como un problema pero no, están correctas, ese char 'A' se traduce a su valor numérico en la tabla ASCII el cual es 65 así haciendo que la operación final sea válida ya que se realiza  $34 * 65$  ( $a * b = 34 * 'A'$ ).

b)

```

#include "stdio.h"
#define EQ(x,y) (x == y)
int main() {
    unsigned int a = -1 ;
    unsigned int b = a << 1 ;
    if (eq(a,b) == EQ(a,b)) {
        printf("Pase con %d y %d", a, b) ;
    }
    return 0 ;
}
int eq(int a, int b) {
    return a == b ;
}

```

En este caso no es tan válido ya que en el main se utiliza el método eq(), pero al estar eq() después del main() provoca un mínimo error de lectura que lo podemos ver en forma de warning si lo compilamos ya que el programa llama a la función antes de que siquiera se defina,

c)

```

#define VARIABLE 777
int main() {
    if (VARIABLE) {
        VARIABLE = 1 ;
    }
    for (;;) {
        return 0 ;
    }
}

```

Aquí se puede ver que VARIABLE se define de manera que se convierta en una constante, así que al momento de que en el "if" tratamos de sustituir la referencia a su valor definido (777) nos da inmediatamente un error así que este caso tampoco es válido.

4) *Escribe un programa en C que calcule el área de una figura geométrica(círculo, triángulo y cuadrilátero, que puede ser cuadrado o rectángulo).El programa se ejecutará como sigue:*

*\$ ./programa <bandera> <datos>+*

*<bandera> indica el tipo de figura geométrica: -c círculo, -t triángulo, -dc cuadrado y -dr rectángulo.*

*<datos>+ corresponde a los elementos necesarios para realizar el cálculo correspondiente.*

*Por ejemplo, si se desea calcular el área de un triángulo el programa debe invocarse así:*

*\$ ./programa -t <base> <altura>*

*Si el programa recibe parámetros incorrectos debe enviar un mensaje indicando la sintaxis correcta.*



**Para desarrollar el programa utiliza directivas del preprocesador, también macros que funcionan como objetos y como funciones. Separa el código de tal forma que en un archivo se encuentren las funciones que implementes, incluyendo a main() y en otro archivo todas las definiciones (macros, bibliotecas de C que utilices y prototipos de funciones).**

Adjunto los archivos de "FigurArea.c" (El archivo donde se encuentra mi función y mi main) y de "MacrosArea.c" (El archivo donde se encuentran las macros) en una carpeta llamada "Pregunta04".

- 5) Escribe un programa en C que imprima el mensaje Hola mundo! Revisa el archivo resultante (el que contiene el programa ejecutable) y reporta su tamaño en bytes. Este archivo es la versión que carga las funciones de biblioteca en forma dinámica. Vuelve a compilar el programa indicando que quieres la versión que carga las bibliotecas en forma estática. Reporta el tamaño del archivo resultante. 2 Explica la razón de la diferencia en los tamaños. Explica por qué crees que el archivo generado tiene el tamaño que reportas.**

Para esta pregunta tuve que hacer 2 intentos, el primero lo hice en windows y aunque lo ejecute como se debía el archivo .out en ambos casos me peso lo mismo (adjunto mi archivo "HolaMundo.c" en una carpeta llamada "Pregunta05"):

```
C:\Users\jamyn>cd C:\Users\jamyn\Documents\Programacion\SistemasOperativos\Tarea01
C:\Users\jamyn\Documents\Programacion\SistemasOperativos\Tarea01>gcc HolaMundo.C -o HolaMundoDin.out
C:\Users\jamyn\Documents\Programacion\SistemasOperativos\Tarea01>.\HolaMundoDin.out
"." no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\Users\jamyn\Documents\Programacion\SistemasOperativos\Tarea01>.\HolaMundoDin.out
Hola mundo
C:\Users\jamyn\Documents\Programacion\SistemasOperativos\Tarea01>gcc -static HolaMundo.C -o HolaMundoSta.out
C:\Users\jamyn\Documents\Programacion\SistemasOperativos\Tarea01>.\HolaMundoSta.out
Hola mundo
```

	HolaMundoDin.out	01/07/2021 06:51 p. m.	Archivo OUT	378 KB
	HolaMundoSta.out	01/07/2021 06:55 p. m.	Archivo OUT	378 KB

Pero en linux, más exactamente en ubuntu, si me dio resultados diferente en ambos casos:

Compilación dinámica:

Tipo:	biblioteca compartida (application/x-shared...
Tamaño:	16.7 kB (16 704 bytes)

Compilación estática:

Tipo:	ejecutable (application/x-executable)
Tamaño:	871.8 kB (871 760 bytes)

La compilación dinámica es mucho más pequeña que la estática, esto es debido a que en la compilación dinámica las librerías se cargan en el momento de ejecución según se van utilizando, por ello el tamaño es mucho más pequeño. Mientras que en la compilación estática las librerías se cargan directamente en el ejecutable haciendo que este sea mucho más pesado.

---

**6) Elige una de las tres siguientes computadoras:**

- a) IBM 360.
- b) VAX 780.
- c) Burroughs 6700.

**Justifica tu elección. Describe las características principales de la arquitectura de la computadora elegida. Describe cual es el sistema operativo que se usaba en esa computadora y cuales son sus características. De los tipos de sistemas operativos vistos en clase, ¿a cual corresponde?**

Elegí la Burroughs 6700, ya que fue de las primeras computadoras en implementar la memoria virtual y que podía soportar el multiprocesamiento, la arquitectura de esta computadora estaba basada en la “arquitectura de pila”, es decir que todos los datos eran guardados en pilas exceptuando los strings, esto hizo que las pilas fueron optimizadas para tener una máxima eficacia y poder lograr la multitarea.

Esta computadora funcionaba con el sistema operativo “Master Control Program” el cual: aceptaba comandos de “input y output”, monitorea y localiza la memoria así como también archivos guardados, reconocía cambios en el sistema, puede iniciar programas y monitorear su rendimiento sin importar si estuvieran instalados en la computadora o estuvieran alojados en discos,

De los sistemas vistos en clase yo diría que pertenece a los monolíticos, esto debido a que se podía enfocar a los procesos de multitarea.

## ***Bibliografía:***

- ❖ <https://es.scribd.com/document/80924630/Diferencias-Entre-C-y-J-ova>
- ❖ [http://www.it.uc3m.es/pbasanta/asng/course\\_notes/ch13s11.html](http://www.it.uc3m.es/pbasanta/asng/course_notes/ch13s11.html)
- ❖ [https://es.wikipedia.org/wiki/Grandes\\_sistemas\\_de\\_Burroughs](https://es.wikipedia.org/wiki/Grandes_sistemas_de_Burroughs)
- ❖ <https://public.support.unisys.com/aseries/docs/ClearPath-MCP-18.0/86000387-512/section-000023488.html>