

Para entregarse el 29/08/2019.

1. Genere un archivo PDF de nombre `boletaInfoAuxiliar.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener dos secciones: una para la información relacionada con el lenguaje **C** y la otra con el lenguaje **Java**. Cada sección debe contener cuatro subsecciones: la primera es para asentar las palabras reservadas; la segunda para los tipos de datos básicos o primitivos con sus longitudes en *bytes*; la tercera para representar en una tabla los operadores con su precedencia y prioridad de evaluación; y la cuarta para los formatos utilizados al escribir información en la salida estándar, para el caso de **C**, los formatos de la función `printf` de `#include<stdio.h>`, los cuales pueden consultarse con el comando `man -a printf` en sistemas unix que tenga instalado un compilador de **C**, para el caso de **Java**, los que se pueden emplear en una “clase” denominada **Formatter**, la cual puede consultarse, por ejemplo, en:

<https://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html>.

2. Genere un archivo PDF de nombre `boletasintaxis.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener la sintaxis para la declaración de tipos de datos básicos o primitivos, de variables de estos tipos de datos, de controles de flujo condicionales y ciclos, así como tipos de datos estructurados y variables de estos tipos en **C** y la descripción de clases y declaración de objetos en **Java**.

Para entregarse el 29/08/2019.

1. Genere un archivo PDF de nombre `boletatipoDBoPyM.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener dos secciones: una para `C` en la que se indique algunas funciones, macros y tipos de datos que pertenezcan al archivo de cabecera `math.h`, `limits.h` y `float.h`; otra para `Java` en la que se indique clases con sus atributos o métodos que permitan el manejo de los tipos de datos primitivos, además de la clase `Math`.
2. Genere un archivo PDF de nombre `boletapalresyop.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener las palabras reservadas y los operadores, para estos últimos junto con su prioridad y precedencia, en `C` y en `Java`.

Para entregarse el 20/09/2019.

1. Genere un archivo PDF de nombre `boletanotacioneval.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener las descripciones de las notaciones infija, prefija o polaca y posfija o polaca inversa. También se debe indicar cómo evaluar una expresión en notación polaca inversa auxiliándose de pilas, así como pasar de la notación infija a la polaca inversa mediante el uso de pilas.
2. Genere un archivo PDF de nombre `boletagenerico.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener una descripción de lo que el alumno entiende por la clase `Object` en `Java` y por variables del tipo `void *` en `C` y del manejo de información en cada uno de esos casos.

Para entregarse el 20/10/2019.

1. Genere un archivo PDF de nombre `boletaarbol.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener la descripción de árbol, árbol binario y árbol binario completo; los recorridos preorden, inorden y posorden; su representación en **C** y **Java**; y pseudocódigo para la generación de un árbol que contenga la derivada simbólica de una expresión contenida en un árbol.
2. Genere un archivo PDF de nombre `boletagrafo.pdf` proveniente de un documento en \LaTeX , donde *boleta* es su número de boleta. El documento debe contener la descripción de grafo y su representación mediante su matriz de adyacencia; también debe describirse la forma de saber si un grafo tiene un recorrido euleriano y, de ser el caso, indicar el recorrido euleriano.

Para entregarse el 20/10/2019.

Definición 1 Suponga que $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$ y que existen $n_0 \in \mathbb{N}$ y $C \in \mathbb{R}^+$ tales que para todo $n \geq n_0$ se cumple $f(n) \leq Cg(n)$, entonces decimos que $f = O(g)$.

1. Muestre que si $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$ y

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \text{alguna constante},$$

entonces $f = O(g)$.

Para cada $f(n)$ en los ejercicios del 2 al 12 indique el inciso que sea una mejor estimación entre las siguientes:

- | | | |
|----------------------|------------------------|------------------------|
| a) $f(n) = O(\ln n)$ | d) $f(n) = O(\ln^2 n)$ | g) $f(n) = O(\ln^3 n)$ |
| b) $f(n) = O(n)$ | e) $f(n) = O(n^2)$ | h) $f(n) = O(n^3)$ |
| c) $f(n) = O(2^n)$ | f) $f(n) = O(n!)$ | i) $f(n) = O(n^n)$ |

2. $\binom{n}{3}$.

3. $10 \ln^3 n + 20n^2$.

4. El número de monomios en x, y, z de grado total a lo más n .

5. El número de polinomios en x de grado a lo más n cuyos coeficientes son 0 o 1.

6. El número de polinomios en x de grado a lo más $n - 1$ cuyos coeficientes son enteros entre 0 o n .

7. El área de una figura fija después de su amplificación en un factor de n .

8. La cantidad de espacio de memoria que una computadora requiere para almacenar el número n .

9. La cantidad de espacio de memoria que una computadora requiere para almacenar n^2 .

10. La suma de los primeros n enteros positivos.

11. La suma de los cuadrados de los primeros n enteros positivos.

12. El número de bits en la suma de los cuadrados de los primeros n enteros positivos.

Para entregarse el 20/10/2019.

Para cada uno de los siguiente numerales, genere un programa que implemente el algoritmo en cuestión e indique qué es lo que se está obteniendo.

Algoritmo 1 Cálculo de s y t

1. **Se require:** Dos enteros positivos a y b con $a > b$.

Se obtiene: s y t tales que $(a \odot s) \oplus (b \odot t) = f(a, b)$

1: $d \leftarrow a$

2: $r \leftarrow b$

3: $s \leftarrow 1$

4: $t \leftarrow 0$

5: $s' \leftarrow 0$

6: $t' \leftarrow 1$

7: **if** $r \neq 0$ **then**

8: $q \leftarrow d \div r$

9:
$$\begin{pmatrix} d & r \\ s & s' \\ t & t' \end{pmatrix} \leftarrow \begin{pmatrix} d & r \\ s & s' \\ t & t' \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix}$$

10: **else**

11: ir al paso 14

12: **end if**

13: ir al paso 3

14: escribir los valores de s y t

Algoritmo 2 Una permutación particular

2. **Se require:** una lista de $n \geq 2$ enteros en un arreglo $A[]$ indexado de 0 a $n - 1$.

Se obtiene: resultado =?

```
1:  $i \leftarrow 0$ 
2:  $j \leftarrow n - 1$ 
3: if  $A[j] < A[j - 1]$  then
4:    $t \leftarrow A[j]$ 
5:    $A[j] \leftarrow A[j - 1]$ 
6:    $A[j - 1] \leftarrow t$ 
7: end if
8:  $j \leftarrow j - 1$ 
9: if  $j > i$  then
10:   ir al paso 3
11: end if
12:  $i \leftarrow i + 1$ 
13: if  $i < n - 1$  then
14:   ir la paso 2
15: end if
```

Para entregarse el 20/10/2019.

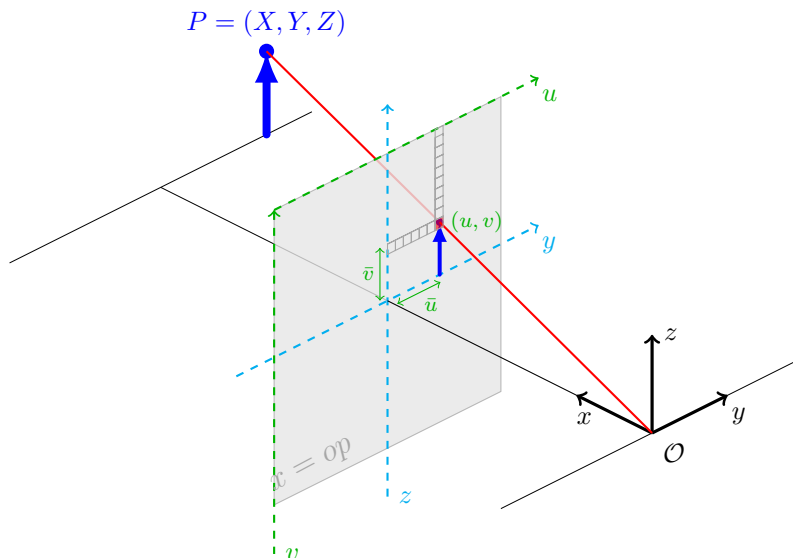
Para cada una de las siguientes tareas, debe generar un documento en L^AT_EX con información de todos y cada uno de los métodos indicados. Seleccione un método de búsqueda y uno de ordenación, genere un programa para representar visualmente la forma en que se desarrolla el método seleccionado y una presentación en dónde se muestre información del método e información del programa realizado.

1. Genere un programa en **C** o en **Java** de nombre *_boleta*busca y la extensión que corresponda. El programa debe mostrar visualmente las comparaciones que se realizan en un arreglo de datos al buscar un elemento en dicha lista. El método de búsqueda a considerar es uno de los siguientes:
 - Búsqueda lineal.
 - Búsqueda binaria.
 - Búsqueda hash.
2. Genere un programa en **C** o en **Java** de nombre *_boleta*ordena y la extensión que corresponda. El programa debe mostrar visualmente las comparaciones que se realizan en un arreglo de datos al ordenar la lista. El método de ordenamiento a considerar es uno de los siguientes:
 - Intercambio directo.
 - Selección directa.
 - Inserción directa.
 - Shellsort.
 - Heapsort.
 - Quicksort.
 - Mergesort.

Para entregarse el 15/11/2019.

En cada una de los siguientes incisos, *boleta* se debe entender como su número de boleta, que será parte del nombre de los programas.

1. Genere un programa en **C** o en **Java** de nombre `_boleta graficaRenR` y la extensión que corresponda, que dada una expresión que represente una función $f : \mathbb{R} \rightarrow \mathbb{R}$, bosqueje su gráfica.
2. Genere un programa en **C** o en **Java** de nombre `_boleta fractales` y la extensión que corresponda, que muestre los fractales descritos en clase.
3. Desarrolle un programa en **C** y en **Java** de nombre `_boleta 3Den2D`, y la extensión que corresponda, que muestre los sólidos platónicos con perspectiva y que se permita rotarlos durante la ejecución del programa.



4. Genere un programa en **C** o en **Java** de nombre `_boleta graficaR2enR` y la extensión que corresponda, que dada una expresión que represente una función $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, bosqueje su gráfica. El bosquejo debe visualizarse empleando la representación indicada en el inciso 3.
5. Desarrolle un programa en **C** y en **Java** de nombre `_boleta 3Den2D`, y la extensión que corresponda, que muestre los sólidos platónicos con perspectiva y que se permita rotarlos durante la ejecución del programa. La representación debe visualizarse empleando la representación indicada en el inciso 3.

6. Para cada uno de las tareas indicadas en los incisos del 3 al 5 desarrolle un programa en **C** y en **Java** de nombre como el correspondiente, pero con sufijo **binocular**, y la extensión que corresponda, que la representación de \mathbb{R}^3 en \mathbb{R}^2 sea como la indicada en el inciso 3, pero empleando un observador con dos ojos en lugar de uno, los ojos se encuentren en el plano xy , el eje x sea el punto medio entre los ojos y que la distancia de cada ojo al eje de las x sea de d unidades (configurable).

Para entregarse el 22/11/2019.

Para cada uno de los siguientes numerales, genere un documento que describa el tema en cuestión, un programa que lo ejemplifique y una presentación en dónde se muestre el tema plasmado en el documento y una pantalla con la ejecución del programa.

1. Compresión RLE y por medio del algoritmo de Huffman.
2. Algoritmos de Bresenham para la recta, la circunferencia y la elipse.
3. Fractales (en una lista circular): Conjuntos de Julia y de Mandelbrot; curvas de Peano, Koch, dragón, Gosper, Hilbert, Lévy; alfombra y triángulo de Sierpiński; y árbol de Pitágoras.
4. Problema de la mochila.
5. Problema de la suma de subconjuntos.
6. Camino Hamiltoniano.
7. Teorema de los cuatro colores.
8. Esquema de cifrado NTRU.
9. Considere tres líneas de transporte tal que para cada línea se conoce la frecuencia y hora inicial de salida de los vehículos de transporte desde cada estación terminal, el tiempo de traslado entre sus estaciones (consideramos que en una misma línea, el tiempo de traslado entre cada estación es el mismo para toda la línea) y el tiempo de transbordo en aquellas estaciones en donde lo haya. Dada una estación inicial y una final, se debe estimar la ruta y el tiempo de traslado que requiera del menor tiempo.
10. Splines: B, cúbicos, de Bézier. Dados la misma cantidad de puntos necesarios, mostrar las diferentes representaciones en una lista circular.
11. Recorrido en un laberinto. Suponga que cuenta con una “cuadrícula” de caracteres de a lo más 50×50 en dónde se representa por una M un muro, por espacio libre acceso, por una E la entrada o lugar de partida en el laberinto y por una o más S la salida del laberinto. Se debe mostrar la ruta que desde la entrada permita llegar a una salida con el menor número de “cuadros” atravesados. Los movimientos solo son horizontales y verticales. Observación, no se permite un movimiento en diagonal (directo, es decir, el equivalente es con dos movimientos si es que no hay muro de pormedio, uno horizontal y otro vertical).

12. Algoritmos de multiplicación: Karatsuba y Schönhage-Strassen. Usando enteros para los factores como de 32 bytes.
13. El problema de los matrimonios estables.
14. El problema de la 3-satisfactibilidad.
15. Juego de ajerez. Al menos dos aperturas y dos defensas con no menos de tres variantes para las primeras cuatro jugadas. Posterior al termino de lo previsto, jugadas al azar, pero dentro de lo reglamentario.
16. Juego de gato para no perder.