

Tarea 12:

Fecha de entrega: **ANTES** del 5 de Noviembre.

Toda la tarea se debe de hacer usando memoria dinámica (`malloc()` y `free()`) cuando sea requerida memoria.

a) Siguiendo con la tarea 20, programar una función que calcula el producto de una matriz por otra matriz (la función debe de verificar que sus dimensiones permiten aplicar la operación). Más información en https://www.varsitytutors.com/hotmath/hotmath_help/spanish/topics/matrix-multiplication

b) Definir una función que consideres complicada para integrar.

- Programar la estimación de la integral en un intervalo $[a,b]$ siguiendo el método aleatorio explicado en esta página https://es.wikipedia.org/wiki/Integraci%C3%B3n_de_Monte_Carlo (en la sección **Algoritmo**),
- Programar la estimación de la integral en un intervalo $[a,b]$ por el método determinista de que está explicado en las primeras 3 figuras de esta página <https://es.khanacademy.org/math/ap-calculus-ab/ab-integration-new/ab-6-3/a/definite-integral-as-the-limit-of-a-riemann-sum>.

c) Hacer el código que da y libera la memoria de un cubo 3D `double *** cubo de dimensiones (tajadas, renglones, columnas)`. De tal forma que se pueda acceder a los elementos del cubo como `cubo[i][j][k]`. En el código, llenar la entradas del cubo con números aleatorios con distribución Normal (con media 0 y varianza 1) siguiendo la siguiente aproximación:

- "Para generar cada número, sume 12 números aleatorios uniformes de 0 a 1 y luego reste 6 a la suma. Esto coincidirá con la media y la desviación estándar de una variable normal. Un inconveniente obvio es que el rango está limitado a ± 6 , a diferencia de una verdadera distribución normal."

Usar archivos de texto para los siguientes incisos. En d) y e) probar con matrices mayores a 10 renglones por 10 columnas.

d) Hacer una función que escribe una matriz (objeto matemático) con valores de punto flotante a un archivo de texto, usar `fprintf()`.

El formato del archivo debe de ser:

```
r c
a11 a12 ... a1c
a21 a22 ... a2c
.
.
.
ar1 ar2 ... arc
```

donde `r` es en número de renglones y `c` el número de columnas.

e) Hacer una función que lee una matriz (objeto matemático) con valores de punto flotante (`float`) de un archivo de texto (con el mismo formato del inciso anterior), usar `fscanf()`.

f) Hacer una función que lee un escenario 2D que está en un archivo de texto de a lo más 24 renglones por 80 columnas (como por ejemplo este https://www.cimat.mx/~alam/elem_comp/mundo_editor.png) y lo guarda en una matriz `char** mundo2D`. Hacer otra función que imprime el escenario 2D en pantalla tomándolo de la matriz `char** mundo2D`. Usar `getc()` como en este código https://www.cimat.mx/~alam/elem_comp/main_text_files.c.

g) Definir una función cualquiera `float fx(float x)` de la cual conozcas o puedas calcular su derivada analítica. Escribe una función que recibe un vector de la evaluación de `fx()` en el intervalo $[a, b]$, con cada evaluación en los puntos $a, a+h, a+2h, \dots, b-h, b$, y que regresa en un vector que se manda por parámetro (al cual se le tiene que asignar la memoria dinámica adentro) la evaluación de la derivada aproximada

por diferencias finitas usando esta fórmula https://www.cimat.mx/~alram/elem_comp/dif_finitas.png que veremos en clase. Finalmente, mide el promedio de las diferencias (error) entre la estimación y la derivada analítica, y reporta el Error Absoluto Medio para todas las evaluaciones (usando la primera fórmula EAM que se muestra aquí https://es.wikipedia.org/wiki/Error_absoluto_medio). ¿el error cambia cuando haces h más pequeño?