Práctica 5: Subalgoritmos

Para cada uno de los enunciados propuestos realizar el análisis del problema, el algoritmo de resolución y la codificación en Lenguaje C, teniendo en cuenta las siguientes etapas:

- Interpretar el enunciado.
- Identificar Datos y Resultados.
- Plantear una Estrategia o Metodología de resolución.
- Escribir el Algoritmo de acuerdo a los puntos anteriores.
- Realizar la traza del algoritmo o prueba de escritorio.
- Codificar el algoritmo en Lenguaje de programación C.
- Verificar el correcto funcionamiento del programa.

1. `POTENCIA`: subalgoritmos que resuelven la operación potencia de un número entero. Para ello defina dos versiones:

- Una función que recibe como parámetros dos números enteros, el primero correspondiente a la base y el segundo al exponente de la potencia a calcular, y que devuelva el resultado.
- Una subrutina con tres parámetros, el primero correspondiente a la base, el segundo al exponente de la potencia a calcular y el tercero donde se guarde el resultado.

Considerar que el exponente puede ser un número entero negativo

2. `VERIFICA`: subalgoritmo que recibe como parámetros a 3 números enteros cualesquiera y que devuelva 1 si el primer parámetro pertenece al intervalo cerrado definido por los otros dos o 0 si no pertenece. No está definido cual de los dos números distintos al primero es el límite inferior del intervalo y cuál es el límite superior.

3. `FACTORIAL`: subalgoritmo que calcula y retorna el factorial de un número entero positivo n mayor o igual a cero, que recibe como parámetro.
Nota: `FACTORIAL(`n`)` se lo escribe matemáticamente como $n!$.

4. `FACTORIALv2`: subalgoritmo que calcula y retorna el factorial de un número entero que recibe como parámetro, salvo que este sea negativo, en ese caso retornará -99.

5. `COMBINATORIO`: subalgoritmo que dados dos números enteros m y n calcula el número combinatorio $\binom{m}{n}$. Recordemos que: $\binom{m}{n} = \frac{m!}{n! \cdot (m-n)!}$.

6. Implementar un algoritmo que dados 3 valores enteros x, y y n:

a. Calcule el valor de $(x+y)^n$

b. Calcule el valor de $\sum_{k=0}^{n} \binom{n}{k} \cdot x^{n-k} \cdot y^k$

Verifique los resultados. Recuerde que: $\sum\limits_{k=0}^{n} \binom{n}{k} \cdot x^{n-k} \cdot y^k = (x+y)^n$

<u>Nota</u>: Para el cálculo de las potencias y de los números combinatorios que necesite invoque adecuadamente a los subalgoritmos `POTENCIA` y `COMBINATORIO`.

7. `PANTALLA`: subalgoritmo que escriba la siguiente sucesión de líneas que se corresponde con opciones de un menú:

```
ACUMULA
LISTA
CONSULTA
FIN
```

y luego permita que el usuario ingrese un carácter (este caracter corresponde a la opción que se quiere ejecutar). La subrutina lo aceptará como válido y lo devolverá sólo si se trata de la primera letra de alguna de las líneas de texto, de lo contrario esperará el ingreso de un nuevo carácter hasta que sea válido para su devolución.

8. Implementar un algoritmo que dados 2 valores enteros m y n:

a. Calcule el valor de $\frac{n!}{(m-1)!}$

b. Calcule el valor de $\prod\limits_{k=m}^{n} k$

Verifique los resultados. Recuerde que: $\prod\limits_{k=m}^{n} k = \frac{n!}{(m-1)!}$

<u>Nota</u>: Para el cálculo de los factoriales que necesite invoque adecuadamente al subalgoritmo `FACTORIAL`.

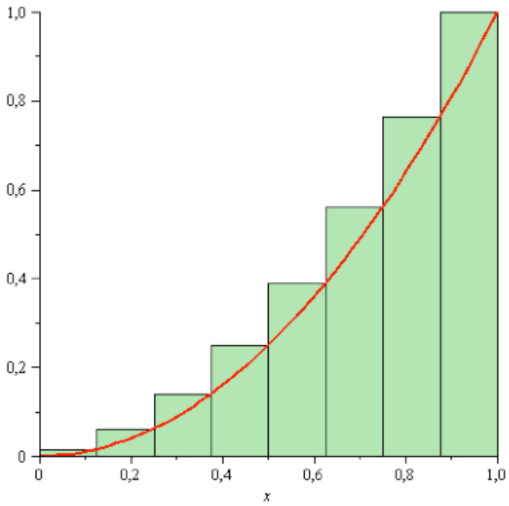
9. `TIEMPO`: subalgoritmo que determina el tiempo transcurrido entre 2 horas distintas del mismo día. Expresar dicho tiempo en cantidad de horas, minutos y segundos.

10. Son conocidas las siguientes fórmulas, las cuales no demostraremos:

- $\sum\limits_{i=1}^{n} i = 1 + 2 + 3 + \ldots + n = n(n+1)/2$
- $\sum\limits_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 \ldots + n^2 = n(n+1)(2n+1)/6$

Para cada igualdad, implemente dos subalgoritmos que permitan verificar que son válidas las fórmulas propuestas para $n = 1, 2, \ldots, 50$.

11. En el siguiente gráfico se muestra una forma de aproximar el área A debajo de la curva $y = x^2$, por arriba de la recta $y = 0$ y entre las rectas $x = 0$ y $x = 1$, a través de la suma de las áreas de 8 rectángulos (R_8) (los rectángulos se obtienen haciendo una partición del intervalo [0,1] en 8 subintervalos iguales)



En forma genérica, recordemos que:

Una forma de aproximar el área de una función real $y = f(x)$ entre dos extremos A y B de su dominio, consiste en:

(I) establecer una partición sobre el intervalo $[A, B]$ mediante un número finito de subintervalos iguales,

(II) para cada subintervalo, trazar un rectángulo cuya altura es el valor de la función $y = f(x)$ en uno de sus extremos (en nuestro caso, el extremo derecho), y

(III) sumar las áreas de todos los rectángulos.

Se pide hacer un algoritmo que permita construir una tabla de valores, para obtener distintas aproximaciones (R_n) del área A, si el intervalo $[0, 1]$ se divide en n subintervalos, para distintos valores de n que el usuario ingresará por teclado, hasta que decida finalizar. Para el cálculo de las aproximaciones (R_n) del área A, crear y utilizar un subalgoritmo función que reciba los parámetros apropiados.

Tema: Sumas de Riemann - https://es.khanacademy.org/math/integral-calculus/riemann-sums-ic