Ciclo 1
Fundamentos de programación con Python
Sesión 20: Fortalecimiento de los conceptos vistos.
Modulo 4

Programa Ciencias de la Computación e Inteligencia Artificial
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda
Bogotá

Agenda

1. Ejercicios de refuerzo listas
2. Ejercicios de refuerzo funciones

1. Ejercicios de refuerzo

El número de combinaciones que podemos formar tomando m elementos de un conjunto con n elementos es:

$$C_n^m = \binom{n}{m} = \frac{n!}{(n-m)!\,m!}.$$

Diseña un programa que pida el valor de n y m y calcule C_n^m. (Ten en cuenta que n ha de ser mayor o igual que m.)
(Puedes comprobar la validez de tu programa introduciendo los valores $n = 15$ y $m = 10$: el resultado es 3003.)

Haz un programa que pida el valor de dos enteros n y m y que muestre por pantalla el valor de

$$\sum_{i=n}^{m} i.$$

1. Ejercicios de refuerzo

Haz un programa que, dados tres valores a, b y c, muestre la función $f(x) = ax^2 + bx + c$ en el intervalo $[z_1, z_2]$, donde z_1 y z_2 son valores proporcionados por el usuario. El programa de dibujo debe calcular el valor máximo y mínimo de $f(x)$ en el intervalo indicado para ajustar el valor de *window_coordinates* de modo que la función se muestre sin recorte alguno.

Define una función que devuelva el número de días que tiene un año determinado. Ten en cuenta que un año es bisiesto si es divisible por 4 y no divisible por 100, excepto si es también divisible por 400, en cuyo caso es bisiesto.

(Ejemplos: El número de días de 2002 es 365: el número 2002 no es divisible por 4, así que no es bisiesto. El año 2004 es bisiesto y tiene 366 días: el número 2004 es divisible por 4, pero no por 100, así que es bisiesto. El año 1900 es divisible por 4, pero no es bisiesto porque es divisible por 100 y no por 400. El año 2000 sí es bisiesto: el número 2000 es divisible por 4 y, aunque es divisible por 100, también lo es por 400.)

2. Ejercicios de refuerzo

Diseña una función *es_primo* que determine si un número es primo (devolviendo *True*) o no (devolviendo *False*). Diseña a continuación un procedimiento *muestra_primos* que reciba un número y muestre por pantalla todos los números primos entre 1 y dicho número.

Vamos a adquirir una vivienda y para eso necesitaremos una hipoteca. La cuota mensual m que hemos de pagar para amortizar una hipoteca de h euros a lo largo de n años a un interés compuesto del i por cien anual se calcula con la fórmula:

$$m = \frac{hr}{1-(1+r)^{-12n}},$$

donde $r = i/(100 \cdot 12)$. Define una función que calcule la cuota (redondeada a dos decimales) dados h, n e i. Utiliza cuantas variables locales consideres oportuno, pero al menos r debe aparecer en la expresión cuyo valor se devuelve y antes debe calcularse y almacenarse en una variable local.

Nota: puedes comprobar la validez de tu función sabiendo que hay que pagar la cantidad de 1 166.75 € al mes para amortizar una hipoteca de 150 000 € en 15 años a un interés del 4.75% anual.

El futuro digital es de todos

MinTIC

Preguntas

UNIVERSIDAD SERGIO ARBOLEDA

MisiónTIC2022