FÍSICA COMPUTACIONAL 1 (DFTE-FIS0610)

COMANDOS BÁSICOS E EXERCÍCIOS PARA O CAPÍTULO 2 - PARTE 1

Alguns comandos básicos em python:

float()
Converte um número para ponto flutuante.
ex: float(2) retorna 2.0

int()
Converte um número para inteiro.
ex: int(2.0) retorna 2

print()
Imprime mensagem na tela
ex: print(”O resultado da primeira operação é”, x, ”e o resultado da segunda é”, y,”.”)

input()
Fornece uma mensagem para o usuário e recebe valores digitados no teclado. Os valores recebidos são do tipo ‘string’, e é preciso convertê-los para o tipo desejado.
ex: x = int(input(”Digite um número inteiro ”))

Operações matemática comuns
Soma (+), subtração (-), produto (*) e divisão (/) possuem os símbolos usuais. Note que a divisão sempre retorna números de ponto flutuante. Outras operações incluem:

//	Parte inteira da divisão de dois números. Ex: 5//2 retorna 2
%	Resto da divisão de dois números. Ex: 5%2 retorna 1
**	Eleva um número a uma dada potência. Ex: 2**4 retorna 16

Exercício 2.1: Operações básicas

Escreva um programa que pede ao usuário dois números, e em seguida fornece:

a) O produto dos dois números

b) A parte inteira e o resto da divisão do primeiro número pelo segundo.

c) O primeiro número elevado ao segundo.

Exercício 2.2: Uma bola que cai de uma torre

Uma bola é lançada de uma torre de altura h, com velocidade inicial zero e é acelerada pela gravidade ($g = 9.81$ m/s^2). Escreva um programa que pede ao usuário a altura da torre em

metros e o tempo em segundos que se passou desde que a bola foi lançada (t), e então calcula a altura atual da bola, ignorando a resistência do ar. Lembre que a altura no tempo t é dada por

$$y = h - \frac{1}{2}gt^2.$$

Importando funções matemáticas:

from math import *

Importa as funções da biblioteca math. Também é possível importar funções individualmente (from math import log,sqrt). Note que as funções desta biblioteca não funcionam com números complexos (a biblioteca cmath trabalha com estes números). Algumas funções comuns:
log: logaritmo de base natural
log10: logaritmo de base 10
exp: exponencial
sin, cos, tan: seno, cosseno, tangente
asin, acos, atan: arco seno, cosseno, tangente
sinh, cosh, tanh: seno, cosseno, tangente hiperbólica
sqrt: raiz quadrada

Exercício 2.3: Convertendo de coordenadas polares para cartesianas

Suponha que a posição de um ponto em um espaço bidimensional nos é dada em coordenadas polares r,θ e queremos converter esta posição para coordenadas cartesianas x,y. Escreva um programa que pede ao usuário para inserir valores de r e θ (em graus), converte os dados para cartesianas ($x = r\cos\theta$, $y = r\sin\theta$) e depois imprime o resultado.

Comparando valores:

Operadores de comparação

Igualdade (==), diferença (! =), maior que (>), menor que (<), maior ou igual que (>=) e menor ou igual que (<=) tem os símbolos esperados. 'and' é o operador lógico 'e'; 'or' é o operador lógico 'ou'.

if()

Um exemplo de uso do 'if' no python é:

```
if (x>10):
    print("O número é maior que 10.")
elif (x>5):
    print("O número é maior que 5.")
else:
    print("O número é menor que 5.")
```

Note que a indentação é usada para indicar ao programa qual parte do código só será executada se a condição for satisfeita.

Exercício 2.4: Determinando se um número é par ou ímpar

Escreva um programa que pede um número ao usuário, determina se este número é par ou ímpar e então imprime se o número é par ou ímpar.

Usando o while:

while()

Um exemplo de uso do 'while' no python é:

```
# Programa que imprime os inteiros positivos menores ou iguais a 10
x = 0
while (x<=10):
    print(x)
    x +=1
```

Note que a primeira linha do programa acima é um comentário.

Exercício 2.5: Série de Fibonacci

Os números de Fibonacci são inteiros em que cada elemento é obtido a partir da soma dos dois últimos, sendo os dois primeiros números 1, 1. Portanto, os primeiros membros da sequência são, 1, 1, 2, 3, 5, 8, 13, 21. Suponha que queremos obter todos os números de Fibonacci menores que 1000. Escreva um programa que nos forneça todos os números desta sequência.

Trabalhando com listas:

range()

Cria um iterador contendo uma lista de valores.
Ex: range(5) retorna a sequência [0,1,2,3,4]
Ex2: range(2,8) retorna a sequência [2,3,4,5,6,7]
Ex3: range(2,20,3) retorna a sequência [2,5,8,11,14,17]

range(início/qntdd, valor lim, passo)

Donde o valor limite está intimamente associada ao passo.

Listas

Permitem armazenar múltiplas variáveis em uma sequência.
Ex: r = [2,4,6] cria uma lista
Ex2: r[0] retorna o primeiro elemento da lista r

Funções que atuam em listas

Considere a lista r = [2,4,6,8,10]. Nos exemplos abaixos, aplicaremos comandos a esta lista.

a) **sum** Soma os elementos de uma lista. sum(r) retorna 30.

b) **max** Retorna o elemento de maior valor de uma lista. max(r) retorna 10

c) **min** Retorna o elemento de menor valor de uma lista. min(r) retorna 2

d) **len** Retorna o número de elementos de uma lista. len(r) retorna 5

e) **r.pop(i)** Retira o elemento i da lista. r.pop(4) remove o 10 da lista r

f) **r.append(n)** Adiciona um número n à lista. r.append(12) adiciona o número 12 a lista r.

map(operacao,lista) Aplica a operação matemática dada a todos os elementos da lista. **ex:**

```
# Programa que calcula a raiz de todos elementos de uma lista
from math import sqrt
r = [4,9,16,25]
raizes = list (map(sqrt,r))
```

Observe que aplicamos a função list() ao resultado de map(). Fizemos isto porque map gera um iterador; aplicando a função list() fazemos com que os elementos do iterador sejam calculados e armazenados na memória.

Exercício 2.6: Trabalhando com listas

Utilize a função range() para gerar uma lista contendo todos os inteiros ímpares menores que 20. Em seguida, determine e imprima:

a) A soma e o número de elementos da lista

b) O maior e o menor elemento da lista

c) O valor médio dos elementos da lista

d) Uma lista contendo o logaritmo de cada membro da lista inicial

---

Problemas opcionais:

Exercício 2.7: Altitude de um satélite

Um satélite deve ser lançado em uma órbita circular em torno da Terra, de modo que ele orbite o planeta uma vez a cada T segundos.

a) Mostre que a altitude h sobre a superfície da Terra que o satélite deve ter é

$$h = \left(\frac{GMT^2}{4\pi^2}\right)^{1/3} - R,$$

onde $G = 6.67 \times 10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$ é a constante gravitacional de Newton, $M = 5.97 \times 10^{24}\,\mathrm{kg}$ é a massa da Terra e $R = 6371\,\mathrm{km}$ é o seu raio.

b) Escreva um programa que pergunta ao usuário o valor desejado de T e então calcula e imprime a altitude correta em metros.

c) Use o seu programa para calcular as altitudes de satélites que orbitam a Terra uma vez por dia (em órbitas chamadas "geoestacionárias"), uma vez a cada 90 minutos e uma vez a cada 45 minutos. O que você concluiu do último destes cálculos? (Um satélite geoestacionário orbita a Terra a uma altitude de ~ 35786 km).

d) Tecnicamente, um satélite geoestacionária é um que orbita a Terra uma vez por *dia sideral*, que corresponde a 23.93 horas, não 24 horas. Qual o motivo disto? E quanto é a diferença que isto causa na altitude de um satélite?

Exercício 2.8: Órbitas planetárias

A órbita no espaço de um objeto em torno de outro, como a de um planeta em torno do Sol, não precisa ser circular. Em geral, ela assume a forma de uma elipse, com o objeto as vezes mais próximo e as vezes mais distante. Se a distância ℓ_1 de maior aproximação de um planeta com relação ao sol, também chamado de *periélio*, e a sua velocidade linear v_1 no periélio forem conhecidas, então qualquer outra propriedade da órbita pode ser calculada a partir destas duas, da seguinte maneira.

a) A segunda lei de Kepler nos diz que a distância ℓ_2 e a velocidade v_2 de um planeta no seu ponto mais distante, ou *afélio*, satisfaz $\ell_2 v_2 = \ell_1 v_1$. Ao mesmo tempo, a energia total, cinética mais gravitacional, de um planeta com velocidade v e distância r do Sol é dada por

$$E = \tfrac{1}{2}mv^2 - G\frac{mM}{r},$$

onde m é a massa do planeta, $M = 1.9891 \times 10^{30}\,\mathrm{kg}$ é a massa do Sol, e $G = 6.6738 \times 10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$ é a constante gravitacional de Newton. Dado que a energia deve ser conservada, mostre que v_2 é a menor raiz da equação quadrática

$$v_2^2 - \frac{2GM}{v_1\ell_1}v_2 - \left[v_1^2 - \frac{2GM}{\ell_1}\right] = 0.$$

Uma vez obtido v_2, nós podemos calcular ℓ_2 usando a relação $\ell_2 = \ell_1 v_1 / v_2$.

b) Dados os valores de v_1, ℓ_1 e ℓ_2, outros parâmetros da órbita podem obtidos usando fórmulas simples que podem ser derivadas das leis de Kepler e do fato que a órbita é

uma elipse.

$$\begin{aligned} \text{Semieixo maior:} \qquad & a = \tfrac{1}{2}(\ell_1 + \ell_2), \\ \text{Semieixo menor:} \qquad & b = \sqrt{\ell_1 \ell_2}\,, \\ \text{Período orbital:} \qquad & T = \frac{2\pi a b}{\ell_1 v_1}\,, \\ \text{Ecentricidade orbital:} \qquad & e = \frac{\ell_2 - \ell_1}{\ell_2 + \ell_1}. \end{aligned}$$

Escreva um programa que pede ao usuário a distância do Sol e a velocidade no periélio, e então calcula e imprime ℓ_2, v_2, T e e.

c) Teste seu programa calculando as propriedades das órbitas da Terra (para a qual $\ell_1 = 1.4710 \times 10^{11}\,\mathrm{m}$ e $v_1 = 3.0287 \times 10^4\,\mathrm{m\,s^{-1}}$) e do cometa Halley ($\ell_1 = 8.7830 \times 10^{10}\,\mathrm{m}$ e $v_1 = 5.4529 \times 10^4\,\mathrm{m\,s^{-1}}$). Entre outras coisas, você deve encontrar que o período orbital da Terra é de um ano e que o do cometa Halley é de cerca de 76 anos.

Exercício 2.9: Números de Catalan

Os números de Catalan C_n são uma sequência de inteiros 1, 1, 2, 5, 14, 42, 132... que desempenham um papel importante em mecânica quântica e na teoria de sistemas desordenados. (Eles foram centrais na prova de Eugene Wigner da chamada lei do Semicírculo.) Eles são dados por

$$C_0 = 1, \qquad C_{n+1} = \frac{4n+2}{n+2}\,C_n.$$

Escreva um programa que imprima em ordem crescente todos os números de Catalan menores que um bilhão.