

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM KHOA CÔNG NGHỆ THÔNG TIN  BỘ MÔN TRÍ TUỆ NHÂN TẠO		<b>ĐỀ THI CUỐI KỲ HỌC KỲ I NĂM HỌC 2021-2022</b>  <b>Môn: NHẬP MÔN LẬP TRÌNH PYTHON</b>  <b>Mã môn học: INPY131685</b>	
Chữ ký giám thị 1	Chữ ký giám thị 2	Đề số/Mã đề: 1	Đề thi có 3 trang
		Ngày thi: 16/01/2022	Thời gian: 90 phút
		Sinh viên được sử dụng tài liệu (kể cả internet)	
<b>Điểm và chữ ký</b>			
CB chấm thi thứ nhất	CB chấm thi thứ hai	<b>Hình thức thi: Làm bài trên máy ở nhà và nộp vào trang dạy học số</b>	

**Chú ý:**

**SV nộp 5 files là Cau1.txt, Cau2.txt, Cau3.txt, Cau4.txt và Cau5.txt vào trang dạy học số**

**Câu 1: (2đ)**

Viết chương trình giải và biện luận phương trình bậc hai  $ax^2 + bx + c = 0$ , nghĩa là nhập  $a, b, c$  và in ra nghiệm.

**Câu 2: (2đ)**

Viết chương trình nhập tháng, năm và in ra tháng đó có bao nhiêu ngày.

**Câu 3: (2đ)**

Viết chương trình nhập vào một số nguyên dương tối đa có 4 chữ số và in ra dạng chữ của số nguyên đó.

VD: 1000 thì in ra một ngàn, 1205 thì in ra một ngàn hai trăm lẻ năm...

**Câu 4: (2đ)**

Viết chương trình mô phỏng quá trình gieo 2 con xúc xắc, mỗi con có 6 mặt, tối thiểu gieo 1000 lần.

Kết quả được trình bày dưới dạng bảng như sau:

Tổng số	Phần trăm mô phỏng
2	2.79
3	5.53
4	8.33
5	11.10
6	13.92
7	16.69
8	13.85
9	11.09

10	8.35
11	5.57
12	2.78

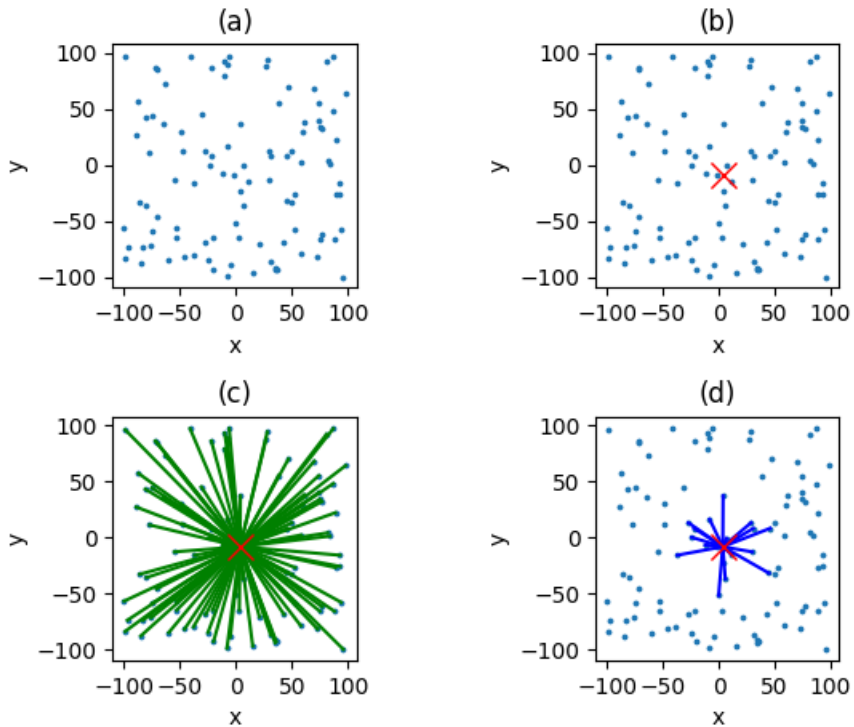
**Câu 5: (2đ)**

Tạo mảng một chiều x gồm 100 số nguyên ngẫu nhiên nằm trong phạm vi [-100, 100]

Tạo mảng một chiều y gồm 100 số nguyên ngẫu nhiên nằm trong phạm vi [-100, 100]

Hướng dẫn: dùng `numpy.random.randint`


- Vẽ các điểm có tọa độ trong x và y như subplot (a)  
 Hướng dẫn: dùng `matplotlib.pyplot.scatter`
- Tính và vẽ thêm điểm trung bình như subplot (b)  
 Hướng dẫn: tính trung bình của mảng dùng `numpy.mean`, vẽ dùng `matplotlib.pyplot.plot`
- Vẽ đoạn thẳng nối điểm trung bình với từng điểm như subplot (c)  
 Hướng dẫn: vẽ dùng `matplotlib.pyplot.plot`
- Vẽ subplot (d) giống như subplot (c) nhưng chỉ vẽ những đoạn thẳng có độ dài nhỏ hơn 50  
 Hướng dẫn: vẽ dùng `matplotlib.pyplot.plot`, độ dài của đoạn thẳng nối 2 điểm  $(x_1, y_1)$  và  $(x_2, y_2)$  là  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$



-----HẾT-----

Ghi chú: Giám thị coi thi không giải thích đề thi

TP. HCM, ngày 15 tháng 01 năm 2022  
**Trưởng Bộ môn**



Trần Tiến Đức

# ĐÁP ÁN CUỐI KỲ HỌC KỲ I NĂM HỌC 2021-2022

## Môn: NHẬP MÔN LẬP TRÌNH PYTHON

### Mã môn học: INPY131685

Ngày thi: 16/01/2022

#### Câu 1: (2đ)

Viết chương trình giải và biện luận phương trình bậc hai  $ax^2 + bx + c = 0$ , nghĩa là nhập a, b, c và in ra nghiệm.

#### Đáp án:

```
import math
a = float(input('Nhập a: '))
b = float(input('Nhập b: '))
c = float(input('Nhập c: '))
if a == 0:
    if b == 0:
        if c == 0:
            print('PTB1 có vô số nghiệm')
        else:
            print('PTB1 vô nghiệm')
    else:
        x = -c/b
        print('PTB1 có nghiệm x = %.2f' % x)
else:
    delta = b**2 - 4*a*c
    if delta < 0:
        print('PTB2 vô nghiệm')
    else:
        x1 = (-b + math.sqrt(delta))/(2*a)
        x2 = (-b - math.sqrt(delta))/(2*a)
```

```
print('PTB2 có nghiệm x1 = %.2f và x2 = %.2f' % (x1, x2))
```

### **Câu 2: (2đ)**

Viết chương trình nhập tháng, năm và in ra tháng đó có bao nhiêu ngày.

```
thang = int(input('Nhap thang: '))
nam = int(input('Nhap nam: '))
print('Thang %d nam %d' % (thang, nam))
if thang in [1, 3, 5, 7, 8, 10, 12]:
    so_ngay = 31
elif thang == 2:
    if (nam % 4) != 0:
        so_ngay = 28
    elif (nam % 100) != 0:
        so_ngay = 29
    elif (nam % 400) != 0:
        so_ngay = 28
    else:
        so_ngay = 29
elif thang in [4, 6, 9, 11]:
    so_ngay = 30
else:
    so_ngay = 0
if so_ngay > 0:
    print('co %d ngay' % so_ngay)
else:
    print('Thang khong hop le')
```

### **Câu 3: (2đ)**

Viết chương trình nhập vào một số nguyên dương tối đa có 4 chữ số và in ra dạng chữ của số nguyên đó.  
VD: 1000 thì in ra một ngàn, 1205 thì in ra một ngàn hai trăm lẻ năm...

**Đáp án:**

```

def mot_chu_so(n):
    chu_so = {0: 'không', 1: 'một', 2: 'hai', 3: 'ba', 4: 'bốn',
              5: 'năm', 6: 'sáu', 7: 'bảy', 8: 'tám', 9: 'chín'}
    }
    result = chu_so[n]
    return result

def hai_chu_so(n):
    chu_so = {0: 'không', 1: 'một', 2: 'hai', 3: 'ba', 4: 'bốn',
              5: 'năm', 6: 'sáu', 7: 'bảy', 8: 'tám', 9: 'chín'}
    }
    chu_so_1 = n // 10
    chu_so_2 = n % 10
    if n == 10:
        return 'mười'
    if n == 11:
        return 'mười một'
    if n == 15:
        return 'mười lăm'
    if chu_so_1 == 1:
        return 'mười ' + chu_so[chu_so_2]
    if chu_so_2 == 0:
        return chu_so[chu_so_1] + ' mươi'
    if chu_so_2 == 1:
        return chu_so[chu_so_1] + ' mươi mốt'
    if chu_so_2 == 5:
        return chu_so[chu_so_1] + ' mươi lăm'
    return chu_so[chu_so_1] + ' mươi ' + chu_so[chu_so_2]

def ba_chu_so(n):
    chu_so = {0: 'không', 1: 'một', 2: 'hai', 3: 'ba', 4: 'bốn',

```

```

        5: 'năm', 6: 'sáu', 7: 'bảy', 8: 'tám', 9: 'chín'
    }
    chu_so_1 = n // 100
    chu_so_2 = (n % 100) // 10
    chu_so_3 = (n % 100) % 10
    so_chuc = n % 100
    if chu_so_2 == 0 and chu_so_3 == 0:
        return chu_so[chu_so_1] + ' trăm'
    if chu_so_2 == 0:
        return chu_so[chu_so_1] + ' trăm lẻ ' + chu_so[chu_so_3]
    return chu_so[chu_so_1] + ' trăm ' + hai_chu_so(so_chuc)
def so_2_chu(n):
    if n < 10:
        return mot_chu_so(n)
    elif n < 100:
        return hai_chu_so(n)
    elif n < 1000:
        return ba_chu_so(n)
    else:
        return 'error: n < 1000'

for n in range(0, 1000):
    print(n, so_2_chu(n))

```

#### **Câu 4: (2đ)**

Viết chương trình mô phỏng quá trình gieo 2 con xúc xắc, mỗi con có 6 mặt, tối thiểu gieo 1000 lần.

#### **Đáp án:**

```

import random

n = 1000000
total_02 = 0

```

```
total_03 = 0
total_04 = 0
total_05 = 0
total_06 = 0
total_07 = 0
total_08 = 0
total_09 = 0
total_10 = 0
total_11 = 0
total_12 = 0
for i in range(0, n):
    mat_1 = random.randint(1, 6)
    mat_2 = random.randint(1, 6)
    sum = mat_1 + mat_2
    if sum == 2:
        total_02 = total_02 + 1
    if sum == 3:
        total_03 = total_03 + 1
    if sum == 4:
        total_04 = total_04 + 1
    if sum == 5:
        total_05 = total_05 + 1
    if sum == 6:
        total_06 = total_06 + 1
    if sum == 7:
        total_07 = total_07 + 1
    if sum == 8:
        total_08 = total_08 + 1
    if sum == 9:
```



```

    total_09 = total_09 + 1
if sum == 10:
    total_10 = total_10 + 1
if sum == 11:
    total_11 = total_11 + 1
if sum == 12:
    total_12 = total_12 + 1
print(' 2 %10.2f' % (total_02/n*100))
print(' 3 %10.2f' % (total_03/n*100))
print(' 4 %10.2f' % (total_04/n*100))
print(' 5 %10.2f' % (total_05/n*100))
print(' 6 %10.2f' % (total_06/n*100))
print(' 7 %10.2f' % (total_07/n*100))
print(' 8 %10.2f' % (total_08/n*100))
print(' 9 %10.2f' % (total_09/n*100))
print('10 %10.2f' % (total_10/n*100))
print('11 %10.2f' % (total_11/n*100))
print('12 %10.2f' % (total_12/n*100))

```

**Câu 5: (2đ)**

Tạo mảng một chiều x gồm 100 số nguyên ngẫu nhiên nằm trong phạm vi [-100, 100]

Tạo mảng một chiều y gồm 100 số nguyên ngẫu nhiên nằm trong phạm vi [-100, 100]

Hướng dẫn: dùng `numpy.random.randint`

a. Vẽ các điểm có tọa độ trong x và y như subplot (a)

Hướng dẫn: dùng `matplotlib.pyplot.scatter`

b. Tính và vẽ thêm điểm trung bình như subplot

(b) Hướng dẫn: tính trung bình của mảng dùng `numpy.mean`, vẽ dùng `matplotlib.pyplot.plot`

c. Vẽ đoạn thẳng nối điểm trung bình với từng điểm như subplot

(c) Hướng dẫn: vẽ dùng `matplotlib.pyplot.plot`

d. Vẽ subplot (d) giống như subplot (c) nhưng chỉ vẽ những đoạn thẳng có độ dài nhỏ hơn 50

**Đáp án:**

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
n = 100
```

```
x = np.random.randint(-100, 100, n)
```

```
y = np.random.randint(-100, 100, n)
```

```
x_mean = np.mean(x)
```

```
y_mean = np.mean(y)
```

```
plt.subplot(2, 2, 1)
```

```
plt.scatter(x, y, s = 3)
```

```
plt.axis([-105, 105, -105, 105])
```

```
plt.axis('equal')
```

```
plt.axis('square')
```

```
plt.xticks([-100, -50, 0, 50, 100])
```

```
plt.yticks([-100, -50, 0, 50, 100])
```

```
plt.xlabel('x')
```

```
plt.ylabel('y')
```

```
plt.title('(a)')
```

```
plt.subplot(2, 2, 2)
```

```
plt.scatter(x, y, s = 3)
```

```
plt.plot(x_mean, y_mean, 'rx', markersize=12)
```

```
plt.axis([-105, 105, -105, 105])
```

```
plt.axis('equal')
```

```
plt.axis('square')
```

```
plt.xticks([-100, -50, 0, 50, 100])
```

```
plt.yticks([-100, -50, 0, 50, 100])
```

```
plt.xlabel('x')
plt.ylabel('y')
plt.title('(b)')
```

```
plt.subplot(2, 2, 3)
plt.scatter(x, y, s = 3)
for i in range(0, n):
    s = np.array([x[i], x_mean])
    t = np.array([y[i], y_mean])
    plt.plot(s,t,'g')
plt.plot(x_mean, y_mean, 'rx', markersize=12)
plt.axis([-105, 105, -105, 105])
plt.axis('equal')
plt.axis('square')
plt.xticks([-100, -50, 0, 50, 100])
plt.yticks([-100, -50, 0, 50, 100])
plt.xlabel('x')
plt.ylabel('y')
plt.title('(c)')
```

```
plt.subplot(2, 2, 4)
plt.scatter(x, y, s = 3)
for i in range(0, n):
    s = np.array([x[i], x_mean])
    t = np.array([y[i], y_mean])
    d = np.sqrt((s[1]- s[0])**2 + (t[1]- t[0])**2)
    if d < 50:
        plt.plot(s,t,'b')
```

```
plt.plot(x_mean, y_mean, 'rx', markersize=12)
plt.axis([-105, 105, -105, 105])
plt.axis('equal')
plt.axis('square')
plt.xticks([-100, -50, 0, 50, 100])
plt.yticks([-100, -50, 0, 50, 100])
plt.xlabel('x')
plt.ylabel('y')
plt.title('(d)')
plt.tight_layout()
plt.show()
----HÉT----
```