

Bài 1:

BÀI TẬP PYTHON CƠ BẢN

1.1. Kiểm tra dữ liệu là number hay string bằng Python

Giả sử viết một chương trình tính tìm số lớn nhất trong 3 số, và yêu cầu người dùng nhập vào ba số đó. Lúc này nếu người dùng nhập vào một chuỗi thì sẽ không đúng, vì vậy ta cần kiểm tra định dạng của nó trước khi thực hiện chương trình.

1.1.1. Kiểu dữ liệu mà Python nhận từ người dùng là gì?

Ở đây chúng ta sử dụng Python 3.

Trong Python, để tiếp nhận dữ liệu từ người dùng thì sử dụng hàm **input**. Có biết hàm này sẽ trả về dữ liệu ở kiểu gì không? Thực ra thì hàm **input** luôn trả về kiểu **string**. Xét ví dụ sau để hiểu rõ hơn:

```
print ("Chương trình Kiểm tra kiểu dữ liệu nhập!")
number = input("Nhập số: ")
string = input("Nhập chuỗi: ")

print("\n")
print("Kiểu dữ liệu vừa nhập là: ")
print("Number ", type(number))
print("String ", type(string))
```

Quan sát kết quả!

Mặc dù biến number nhập vào là số 12 nhưng nó vẫn là kiểu string.

1.1.2. Cách kiểm tra dữ liệu là string hay number trong Python

Trong Python, có thể sử dụng nhiều cách để kiểm tra dữ liệu input là chuỗi hay số.

Cách 1: Chuyển đổi thành int hoặc float để kiểm tra input là số hay chuỗi.

Với cách này thì ta sử dụng hai hàm **int()** và **float()** để ép kiểu dữ liệu, dựa vào kết quả trả về ta sẽ biết chính xác người dùng nhập vào một chuỗi hay một số.

Ta phải đặt nó trong lệnh try .. except, bởi hai hàm ép kiểu trên sẽ không thể ép một chuỗi thành một số trong trường hợp giá trị của chuỗi không phải là số. Ví dụ chuyển chuỗi "Python" thành số là hoàn toàn sai.

Quá trình kiểm tra như sau: Nếu ép kiểu thành công thì là số, ngược lại thì một Exception sẽ được bung ra nên nó là một chuỗi.

```
print ("Chương trình Kiểm tra kiểu dữ liệu nhập!")
user_input = input("Nhập một số: ")
```

```
print("\n")
try:
    val = int(user_input)
    print("Dữ liệu nhập vào là int = ", val)
except ValueError:
    try:
        val = float(user_input)
        print("Dữ liệu nhập vào là float = ", val)
    except ValueError:
        print("Dữ liệu nhập vào không phải là number!")
```

Cách 2: Sử dụng hàm isdigit để kiểm tra là number hay string.

```
print ("Chương trình Kiểm tra kiểu dữ liệu nhập!")
num = input("Nhập số: ")

print("\n")
if num.isdigit():
    print("Dữ liệu nhập vào là số!")
else:
    print("Dữ liệu nhập vào là chuỗi!")
```

Tuy nhiên hàm này chỉ hoạt động với số nguyên dương. Nếu nhập vào là một số thực hoặc một số âm thì nó sẽ trả về là chuỗi. Vì vậy, giải pháp tốt nhất vẫn là chọn cách 1.

1.1.3. **Viết chương trình nhập vào một số bằng Python**

Bây giờ ta sẽ viết chương trình ép buộc người dùng nhập vào là một số, nếu nhập vào là một chuỗi thì yêu cầu nhập lại.

```
while True:
    num = input("Vui lòng nhập vào một số!")
    try:
        val = int(num)
        print("Dữ liệu nhập vào là số!")
        print("Giá trị của nó là: ", val)
        break; # Thoát khỏi vòng lặp
    except ValueError:
        try:
            float(num)
            print("Dữ liệu nhập vào là float!")
```

```

        print("Giá trị của nó là: ", val)
        break; # Thoát khỏi vòng lặp
    except ValueError:
        print("Đây không phải là một số, vui lòng nhập lại!")

```

Chạy chương trình và thử nhập vào một chuỗi xem kết quả!

1.2. Tính tổng $S(n) = 1 + 2 + 3 + \dots + n$ bằng Python

Hãy sử dụng kiến thức vòng lặp while hoặc vòng lặp for của Python để giải bài tập sau:

Tính tổng $S(n) = 1 + 2 + 3 + \dots + n$.

Đây là chương trình tính tổng từ 1 đến n, với n được nhập từ bàn phím, sử dụng ngôn ngữ Python để giải.

Bài giải

Với bài này thì ta chỉ cần sử dụng hai thao tác như sau:

- ✎ Tạo một biến tong để lưu trữ tổng của dãy số đó.
- ✎ Sử dụng vòng lặp để lặp từ 1 đến n.

Ta sẽ sử dụng hàm input để yêu cầu người dùng nhập vào số n.

Sử dụng vòng lặp for:

```

tong = 0
n = 1

print("Tính tổng S(n) = 1 + 2 + 3 + ... + n")

# Nhập dữ liệu
print("Hãy nhập vào số n: ")
n = int(input())

# Tính tổng
for i in range(0, n+1):
    tong += i

# In kết quả
print("Tổng là: ", tong)

```

Sử dụng vòng lặp while:

```

tong = 0
n = 1
i = 1

```

```
print("Tính tổng S(n) = 1 + 2 + 3 + ... + n")
```

```
# Nhập dữ liệu
```

```
print("hãy nhập vào số n: ")
```

```
n = int(input())
```

```
# Tính tổng
```

```
while i <= n :
```

```
    tong += i
```

```
    i += 1
```

```
# In kết quả
```

```
print ("Tổng là: ", tong)
```

1.3. Tính tổng $S(n) = 1^2 + 2^2 + \dots + n^2$ bằng Python

Viết chương trình tính tổng sau bằng cách sử dụng ngôn ngữ Python.

Tính $S(n) = 1^2 + 2^2 + \dots + n^2$

Gợi ý: Sử dụng vòng lặp và toán tử số mũ để giải.

- ✎ Trong Python có toán tử tính số mũ nên ta có thể áp dụng trong bài này. Ví dụ 2^3 thì sẽ là `2 ** 3`.
- ✎ Ta sẽ sử dụng vòng lặp for trong Python để lặp qua các phần tử từ 1 đến n.
- ✎ Sử dụng `range(1, n+1)` để tạo ra một dãy số từ 1 đến n.

Để chương trình đơn giản hóa lại thì mình bỏ qua phần kiểm tra dữ liệu nhập vào phải là số không nhé.

Bài giải như sau:

```
# BAI TAP PYTHON CO BAN
```

```
# TAC GIA: NGUYEN VAN TEO
```

```
# Tính S(n) = 1^2 + 2^2 + ... + n^2
```

```
# NGAY: 07/02/2023
```

```
print("Chương trình Tính S(n) = 1^2 + 2^2 + ... + n^2!")
```

```
tong = 0
```

```
n = 0
```

```
print("Hãy nhập vào số n: ")
```

```
n = int(input())

for i in range(1, n + 1) :
    tong += i ** 2

print("Tổng số là: ", tong)
```

Chạy chương trình và xem kết quả!

1.4. Tính tổng $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ bằng Python

Sử dụng ngôn ngữ Python cơ bản giải bài tập sau:

Tính $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

Gợi ý:

Với bài này thì ta phải lưu trữ dữ liệu kiểu float, vì phép chia có thể trả về giá trị số lẻ có phần dư.

Tương tự, sử dụng vòng lặp để lặp qua dãy số từ 1 đến n, tức khoảng range sẽ được tính là range(1, n+1).

Bài giải như sau:

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Tính tổng  $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$  bằng Python
# NGÀY: 07/02/2023

print("Chương trình tổng  $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$  bằng Python!")

tong = 0
n = 0

print("Hãy nhập vào số n: ")
n = int(input())

for i in range(1, n + 1) :
    tong += 1 / i

print("Tổng số là: ", tong)
```

Chạy chương trình và xem kết quả!

1.5. Tính tổng $S(n) = \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2n}$ bằng Python

Sử dụng ngôn ngữ Python để giải bài toán tính tổng sau:

Tính $S(n) = \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2n}$

Gợi ý:

✍️ Áp dụng công thức $\frac{1}{2n}$ để giải cho bài toán này.

Bài giải như sau:

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Tính S(n) = 1/2 + 1/4 + ... + 1/2n
# NGÀY: 07/02/2023
print("Chương trình Tính S(n) = 1/2 + 1/4 + ... + 1/2n!")

tong = 0
n = 0

print("Hãy nhập vào số n: ")
n = int(input())

for i in range(1, n + 1) :
    tong += 1 / (2 * i)

print("Tổng số là: ", tong)
```

Chạy chương trình và xem kết quả!

1.6. Tính **tổng** $S(n) = \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{(2n + 1)}$

Sử dụng ngôn ngữ Python để thực hiện bài toán sau đây.

Tính $S(n) = \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{(2n + 1)}$

Như vậy, công thức chung cho mỗi phần tử đó là $\frac{1}{(2n + 1)}$.

Để làm bài tập Python cơ bản này thì ta sử dụng vòng lặp for, kết hợp với công thức $\frac{1}{(2n + 1)}$ sẽ đưa ra được thuật toán như sau.

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Tính S(n) = 1/3 + 1/5 + ... + 1/(2n + 1)
# NGÀY: 07/02/2023

print("Chương trình Tính S(n) = 1/3 + 1/5 + ... + 1/(2n + 1)")

tong = 0
```

```

n = 0

print("Hãy nhập vào số n: ")
n = int(input())

for i in range(1, n + 1):
    tong += 1 / (2 * i + 1)

print("Tổng số là: ", tong)

```

Chạy chương trình và xem kết quả!

1.7. Liệt kê tất cả các ước số của số nguyên dương n bằng Python

Trong bài này ta sẽ viết chương trình Python liệt kê tất cả các ước số của N với N được nhập vào từ bàn phím.

Ví dụ nhập vào số N = 12, thì kết quả sẽ trả về các số 1, 2, 3, 4 và 12.

Gợi ý:

Để giải bài này bằng ngôn ngữ Python thì ta thực hiện theo các bước sau.

- ✎ Bước 1. Yêu cầu nhập vào số N
- ✎ Bước 2: Sử dụng vòng lặp để lặp từ 1 đến N, nếu số nào mà N chia hết thì đó chính là ước.

```

# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Chương trình Python liệt kê tất cả các ước số của N
# NGÀY: 07/02/2023

print("Chương trình Python liệt kê tất cả các ước số của N")

print("Nhập vào số N cần tìm ước: ")

n = int(input())
print("Danh sách các ước số của ", n, " là")
for i in range(1, n+1):
    if (n % i == 0):
        print(i, end=", ")

```

Chạy chương trình và xem kết quả!

1.8. Tính tổng tất cả các ước số của số nguyên dương N bằng Python

Hãy viết chương trình tính tổng tất cả các ước số của số nguyên dương N bằng Python, điều kiện là N được nhập từ bàn phím.

Gợi ý:

Bài này có cách giải tương tự như bài trước, có điều thay vì liệt kê thì ta khai báo thêm một biến tổng để lưu trữ tổng số trong vòng lặp.

Sau đây là chương trình được giải bằng ngôn ngữ lập trình Python.

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Tính tổng tất cả các ước số của số nguyên dương N bằng Python!
# NGÀY: 07/02/2023

print("Chương trình Tính tổng tất cả các ước số của số nguyên dương N")

print("Nhập vào số N cần tính tổng các ước số: ")

n = int(input())
sum = 0

for i in range(1, n+1):
    if (n % i == 0):
        sum += i

print("Tổng tất cả các ước số ", n, " là: ", sum)
```

Chạy chương trình và xem kết quả!

1.9. Tìm ước số lẻ lớn nhất của một số bằng Python

Hãy viết chương trình tìm ước số lẻ lớn nhất của một số bằng ngôn ngữ Python. Ví dụ ta nhập số 50 thì ước số lẻ lớn nhất của nó sẽ là 25, nhập số 51 thì ước số lẻ lớn nhất là 51.

Gợi ý:

Để giải bài này trong Python thì ta thực hiện theo các bước như sau.

- ✍ **Bước 1:** Yêu cầu người dùng nhập vào số N bằng hàm input.
- ✍ **Bước 2:** Lặp từ N trở về 1, số lẻ đầu tiên chia hết cho N chính là kết quả cần tìm.

Bài giải như sau:

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Tìm ước số lẻ lớn nhất của một số bằng Python!
# NGÀY: 07/02/2023

print("Chương trình Tìm ước số lẻ lớn nhất của một số bằng Python!")

print("Nhập vào số N: ")

n = int(input())
result = 0

for i in range(n+1, 1, -1):
    if (n % i == 0 and i % 2 != 0):
        result = i
        break

print("Ước số lẻ lớn nhất của ", n, " là: ", result)
```

Chạy chương trình và xem kết quả!

1.10. Kiểm tra một số có phải là số hoàn hảo bằng Python

Hãy viết chương trình kiểm tra một số có phải là số hoàn hảo không bằng Python, đây là bài tập cơ bản giúp bạn luyện tư duy lập trình Python.

Gợi ý:

Để giải bài này thì trước tiên bạn phải tìm hiểu một chút về khái niệm số hoàn hảo.

Gọi N là một số hoàn hảo thì: Tổng các ước từ 1 -> (N - 1) sẽ bằng N.

Ví dụ: 6 là số hoàn hảo vì tổng các ước từ 1 đến 5 là $(1 + 2 + 3) = 6$.

Như vậy với bài này ta sẽ giải bằng ngôn ngữ Python như sau:

- ✎ Bước 1: Yêu cầu người dùng nhập số cần kiểm tra bằng hàm input()
- ✎ Bước 2: Lặp từ 1 đến N - 1, tính tổng tất cả các ước số của N trong khoảng đó.
- ✎ Bước 3: Nếu tổng bằng N thì N chính là số hoàn hảo.

Bài giải như sau:

```
# BAI TAP PYTHON CO BAN
```

```
# TAC GIA: NGUYEN VAN TEO
# Kiểm tra một số có phải là số hoàn hảo bằng Python
# NGÀY: 07/02/2023

print("Chương trình Kiểm tra một số có phải là số hoàn hảo bằng Python!")

print("Nhập vào số N lớn hơn 0: ")

n = int(input())
tong = 0

for i in range(1, n):
    if (n % i == 0):
        tong += i

if (tong == n):
    print(n, " là số hoàn hảo")
else:
    print(n, " không phải là số hoàn hảo")
```

Chạy chương trình và xem kết quả!

1.11. Kiểm tra số chính phương bằng Python

Hãy viết chương trình kiểm tra số chính phương bằng Python, qua bài này sẽ giúp hiểu được số chính phương là gì, cũng như thuật toán để kiểm tra số chính phương bằng Python.

Gợi ý:

Số chính phương là số có giá trị bằng tổng bình phương của một số nguyên nào đó. Ví dụ số 4 là số chính phương vì nó $4 = 2^2$.

Để giải bài này thì chỉ cần kiểm tra trong phạm vi từ 1 đến N xem có số nào bình phương lên sẽ bằng N hay không. Nếu có thì N là số chính phương, ngược lại thì không phải.

Ta sẽ sử dụng vòng lặp Python để giải bài này như sau:

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Kiểm tra số chính phương bằng Python.
# NGÀY: 07/02/2023
```

```

print("Chương trình Kiểm tra số chính phương bằng Python!")

print("Nhập vào số N lớn hơn 0: ")

n = int(input())
check = False

for i in range(1, n + 1):
    if (i**2 == n):
        check = True
        break

if (check == True):
    print(n, " là số chính phương")
else:
    print(n, " không phải là số chính phương")

```

Nhận xét: Thực ra với bài này ta không cần phải lặp từ 1 đến N mà chỉ cần lặp từ 1 đến $(N/2 + 1)$, bởi vì thực tế những số lớn hơn $(N/2 + 1)$ bình phương lên sẽ luôn luôn lớn hơn N.

1.12. Kiểm tra số nguyên tố bằng Python

Hãy viết chương trình kiểm tra số nguyên tố bằng Python.

Gợi ý:

Số nguyên tố là số nguyên lớn hơn 1 và chỉ chia hết cho 1 và chính nó.

Dựa vào định nghĩa này ta rút ra được những tính chất sau: Số 2 là SNT chẵn duy nhất, bởi vì những số chẵn lớn hơn 2 luôn chia hết cho 1, 2 và chính nó.

Nếu một số thỏa 2 tính chất trên thì đó là số nguyên tố.

Dựa vào đây ta có thể phân tích toán như sau: Giả sử N là số cần kiểm tra.

- ✂ Kiểm tra nếu N bằng 2 thì là SNT => Break
- ✂ Nếu $N \leq 1$ thì không phải SNT => Break
- ✂ Nếu N là số chẵn thì không phải SNT => Break
- ✂ Lặp qua các số lẻ từ 2 -> $(N - 1)$, nếu tồn tại số nào mà N chia hết thì không phải là SNT, nếu không thì đó là SNT.

Bài giải viết bằng ngôn ngữ Python như sau:

```

# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Kiểm tra số nguyên tố bằng Python!

```

```
# NGÀY: 07/02/2023

print("Chương trình Kiểm tra số nguyên tố bằng Python!")

print("Nhập vào số N lớn hơn 1: ")

# Lấy dữ liệu
n = int(input())
flag = True

# Kiểm tra SNT
if (n < 2):
    flag = False
elif (n == 2):
    flag = True
elif (n % 2 == 0):
    flag = False
else:
    # Lặp qua các số lẻ nên bắt đầu từ 3 với bước nhảy là 2
    for i in range(3, n, 2):
        if (n % i == 0):
            flag = False

# In kết quả
if flag == True:
    print(n, " là số nguyên tố!")
else:
    print(n, " không phải là số nguyên tố!")
```

Chạy chương trình và quan sát kết quả. Hãy tối ưu chương trình trên.

1.13. Đảo ngược một số bằng Python

Viết chương trình đảo ngược một số bằng Python. Ví dụ có số 12345 thì đảo ngược nó thành 54321 bằng cách sử dụng ngôn ngữ lập trình Python.

Thuật toán giải bài này bằng Python khá đơn giản, ta chỉ cần chia lấy dư số đó cho 10, phần dư chính là số đảo đầu tiên, tiếp theo lấy phần nguyên chia tiếp cho 10 và cứ như thế cho tới khi phần nguyên bằng 0.

Cách giải 1: Chỉ in kết quả lên màn hình.

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# ĐẢO NGƯỢC MỘT SỐ BẰNG PYTHON
# NGÀY: 07/02/2023

print("Chương trình Đảo ngược một số bằng Python!")

print("Nhập vào số N cần đảo ngược: ")

# Lấy dữ liệu
n = int(input())

# In ra số đảo ngược
while (n != 0):
    print(n % 10, end="")
    n = n // 10 # Chia lấy phần nguyên
```

Chạy chương trình và quan sát kết quả.

Cách giải 2: Trả kết quả về một biến và in lên màn hình.

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# ĐẢO NGƯỢC MỘT SỐ BẰNG PYTHON
# NGÀY: 07/02/2023

print("Chương trình Đảo ngược một số bằng Python!")

print("Nhập vào số N cần đảo ngược: ")

# Lấy dữ liệu
n = int(input())
result = ""

# In ra số đảo ngược
while (n != 0):
    result += str(n % 10)
    n = n // 10 # Chia lấy phần nguyên

print(result)
```

Chạy chương trình và quan sát kết quả.

1.14. In ra từng ký tự của một số bằng Python

Viết chương trình in ra từng ký tự của một số bằng Python. Ví dụ ta có số 12345 thì hãy in lần lượt các số 1 2 3 4 5 lên màn hình console.

Để giải bài này thì ta có thể thông qua hai bước như sau:

- ✎ Bước 1: Đảo ngược số cần in đó, ví dụ từ 12345 thành 54321.
- ✎ Bước 2: In lần lượt các ký tự của số đã đảo đó ra màn hình.

Bài giải như sau:

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# IN RA TUNG KY TU CUA MOT SO BANG PYTHON
# NGAY: 07/02/2023

print("Chương trình In ra từng ký tự của một số bằng Python!")

print("Nhập vào số N: ")

# Lấy dữ liệu
n = int(input())
so_dao_nguoc = ""

# B1: Đảo ngược số cần in ra
while (n != 0):
    so_dao_nguoc += str(n % 10)
    n = n // 10 # Chia lấy phần nguyên

# B2: In lần lượt các ký tự từ cuối đến đầu của số đã đảo
so_dao_nguoc = int(so_dao_nguoc, 10) # Đổi string sang int

while (so_dao_nguoc != 0):
    print(so_dao_nguoc % 10, end= ' - ')
    so_dao_nguoc = so_dao_nguoc // 10 # Chia lấy phần nguyên
```

Chạy chương trình và quan sát kết quả.

1.15. Giải phương trình bậc nhất một ẩn bằng Python ($ax + b = 0$)

Hãy viết chương trình giải phương trình bậc nhất một ẩn bằng Python ($ax + b = 0$).

Gợi ý:

Giả sử ta có phương trình $ax + b = 0$, đây là phương trình chỉ có một nghiệm duy nhất đó là: $x = -b/a$.

Vậy các bước để giải nó trong Python như sau:

- ✂ Bước 1: Yêu cầu nhập vào hai số a và b
- ✂ Bước 2: Kiểm tra nếu $a = 0$ thì yêu cầu nhập lại a, vì phương trình này có điều kiện $a \neq 0$.
- ✂ Bước 2: Tìm nghiệm bằng công thức $x = -b/a$

Bài giải như sau:

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# Giai PT bac 1: ax + b = 0
# NGAY: 07/02/2023

print("Chương trình Giai PT bac 1: ax + b = 0!")

# Nhập số a và kiểm tra điều kiện a khác 0
print("Nhập vào số a: ")
a = int(input())

while True:
    if a == 0:
        print("Vui lòng nhập số a khác 0: ")
        a = int(input())
    else:
        break

# Nhập số b
print("Nhập vào số b: ")
b = int(input())

# Nghiệm
print("Nghiệm của phương trình là x = ", (-b / a))
```

Chạy chương trình và nếu bạn nhập $a = 0$ thì Python sẽ nhắc bạn nhập lại.

1.16. Giải phương trình bậc hai một ẩn bằng Python ($ax^2 + bx + c = 0$)

Hãy viết chương trình giải phương trình bậc hai một ẩn $ax^2 + bx + c = 0$ bằng Python.

Bài giải:

Để giải phương trình bậc hai $ax^2 + bx + c = 0$ thì ta phải tính chỉ số Delta, dựa vào Delta này mà ta xác định được nghiệm. Cụ thể như sau:

Bước 1: Tính $\Delta = b^2 - 4ac$

Bước 2: Sau khi tính được Delta thì ta sẽ có một số trường hợp như sau:

- Nếu $\Delta < 0$ thì phương trình vô nghiệm
- Nếu $\Delta = 0$ thì phương trình có nghiệm kép $x_1 = x_2 = -\frac{b}{2a}$
- Nếu $\Delta > 0$ thì phương trình có hai nghiệm phân biệt $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$

Viết chương trình trong Python như sau:

- ✎ Bước 1: Nhập các số a, b và c. Kiểm tra điều kiện một trong hai số a và b phải khác 0.
- ✎ Bước 2: Tính Delta.
- ✎ Bước 3: Dựa vào Delta để tìm ra nghiệm của phương trình (có 3 trường hợp).

```
# BAI TAP PYTHON CO BAN
# TAC GIA: NGUYEN VAN TEO
# GIAI PHUONG TRINH BAC HAI 1 AN  $ax^2 + bx + c = 0$ 
# NGAY: 07/02/2023

import math

print("Chương trình Giai PT bac 2:  $ax^2 + bx + c = 0$ !")

# Nhập số a và kiểm tra điều kiện khác 0
print("Nhập vào số a: ")
a = int(input())

print("Nhập vào số b: ")
b = int(input())
```



```

while True:
    if a == 0 and b == 0:
        print("Một trong hai số a và b phải khác 0: ")
        print("Nhập lại số a: ")
        a = int(input())

        print("Nhập lại số b: ")
        b = int(input())
    else:
        break

# Nhập số c
print("Nhập vào số c: ")
c = int(input())

# Tính Delta
delta = b**2 - 4 * a * c

# Nghiệm
if delta < 0:
    print("Phương trình vô nghiệm!")
elif delta == 0:
    print("Phương trình có nghiệm kép x1 = x2 = ", -(b / (2 * a)) )
else:
    print("Phương trình có hai nghiệm phân biệt:")
    print("x1 = ", (-(b) + math.sqrt(delta))/(2*a) )
    print("x1 = ", (-(b) - math.sqrt(delta))/(2*a) )

```

Chạy chương trình và quan sát kết quả.