

```
In [1]: a=int(input("Enter a num a: "))
        b=int(input("Enter a num b: "))
        print(type(a))
        print(type(b))
```

```
Enter a num a: 2
Enter a num b: 2
<class 'int'>
<class 'int'>
```

```
In [4]: a=input("Enter a num a: ")
        b=input("Enter a num b: ")
        print(type(a))
        print(type(b))
```

```
Enter a num a: 2
Enter a num b: 2
<class 'str'>
<class 'str'>
```

```
In [8]: #Assigning a variable
        a = "siri"
        b = "sree"
        c = "srinitha"
        print(a+" "+b+" "+c)
```

```
siri sree srinitha
```

```
In [13]: #taking a input from user
        a=input("Enter a value: ")
        print("the value of a is:"+a)
```

```
Enter a value: siri
the value of a is:siri
```

```
In [14]: #taking a input from user
        a=input("Enter a value: ")
        print("the value of a is:"+a)
```

```
Enter a value: 3
the value of a is:3
```

```
In [15]: #taking a input from user
        a=int(input("Enter a value: "))
        print("the value of a is:"+a)
```

```
Enter a value: siri
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[15], line 2
      1 #taking a input from user
----> 2 a=int(input("Enter a value: "))
      3 print("the value of a is:"+a)

ValueError: invalid literal for int() with base 10: 'siri'
```

```
In [18]: #taking a input from user
        a=int(input("Enter a value: "))
        print("the value of a is "+str(a))
```

```
Enter a value: 3
the value of a is 3
```

1. Print your name.
2. Print the result of adding two numbers.
3. Print the result of subtracting two numbers.
4. Print the result of multiplying two numbers.
5. Print the result of dividing two numbers.

In [19]: `print("siri")`

siri

In [20]: `a=10
b=20
print(a+b)`

30

In [21]: `a=input("Enter a value of a: ")
b=input("Enter a value of b: ")
print(a+b)`

Enter a value of a: 20
Enter a value of b: 20
2020

In [22]: `a=int(input("Enter a value of a: "))
b=int(input("Enter a value of b: "))
print(a+b)`

Enter a value of a: 20
Enter a value of b: 20
40

In [23]: `a=input("Enter a value of a: ")
b=input("Enter a value of b: ")
print(a-b)`

Enter a value of a: 20
Enter a value of b: 20

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[23], line 3  
      1 a=input("Enter a value of a: ")  
      2 b=input("Enter a value of b: ")  
----> 3 print(a-b)
```

TypeError: unsupported operand type(s) for -: 'str' and 'str'

In [24]: `a=int(input("Enter a value of a: "))
b=int(input("Enter a value of b: "))
print(a-b)`

Enter a value of a: 20
Enter a value of b: 20
0

In [26]: `a=int(input("Enter a value of a: "))
b=int(input("Enter a value of b: "))
print(a*b)`

Enter a value of a: 2
Enter a value of b: 3
6

```
In [27]: a=int(input("Enter a value of a: "))
b=int(input("Enter a value of b: "))
print(a/b)
```

```
Enter a value of a: 2
Enter a value of b: 3
0.6666666666666666
```

```
In [28]: #DAY 2
```

```
In [29]: #varibales
my_int = 20
print(my_int)
```

```
20
```

1. Declare two variables `a` and `b`, assign integer values to them, and print their sum.

- **Expected Output:** The sum of `a` and `b`.

```
In [1]: a=10
b=20
sum=a+b
print(sum)
```

```
30
```

2. Create a variable `name` and assign your name to it. Print a greeting message using your name.

- **Expected Output:** Greeting message with your name, e.g., "Hello, John!"

```
In [9]: variable="hello"
print(variable + " siri ")
```

```
hello siri
```

3. Define a variable `pi` and assign the value of π (pi) to it. Print the value of `pi`.

- **Expected Output:** The value of π (pi), e.g., 3.14159.

```
In [19]: pi=3.14
print(" the value of pi is "+ pi)#here direct it does not concatenation of str or float
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[19], line 2
      1 pi=3.14
----> 2 print(" the value of pi is "+ pi)

TypeError: can only concatenate str (not "float") to str
```

```
In [18]: pi=3.14
print(" the value of pi is "+ str(pi))
```

```
the value of pi is 3.14
```

4. Define a variable `is_raining` and ask the user to input either "True" or "False" (as a string). Convert the input to a boolean and print its type.

- **Expected Input:** "True" or "False"

- **Expected Output:** The data type of the converted boolean.

```
In [6]: is_raining=input("Either True or False")
is_raining=is_raining==True
print(type(is_raining))
```

```
Either True or FalseTrue
<class 'bool'>
```

5. Create a string variable `sentence` containing any sentence of your choice. Ask the user to input a number, convert it to an integer, and print the sentence repeated that number of times.

- **Expected Input:** A number (e.g., "3")
- **Expected Output:** The sentence repeated the specified number of times.

```
In [9]: sentence="The python is good"
num=input("Enter a num: ")
#Cov to int
num = int(num)
#print the sentence repeated that number of times.
print(sentence*num)
```

```
Enter a num: 3
The python is goodThe python is goodThe python is good
```

6. Given two variables `x` and `y`, perform the following operations and print the results:

- Addition of `x` and `y`.
- Subtraction of `y` from `x`.
- Multiplication of `x` and `y`.
- Division of `x` by `y`.
- `x` raised to the power of `y`.
- The remainder when `x` is divided by `y`.
- The floor division of `x` by `y`.

```
In [11]: x=int(input("Enter the value of x: "))
y=int(input("Enter the value of y: "))
print("Addition of x and y", x+y)
print("Subtraction of y from x",x-y)
print("Multiplication of x and y",x*y)
print("Division of x by y",x/y)
print("x raised to the power of y",x**y)
print("The remainder when x is divided by y",x%y)
print("The floor division of x by y.",x//y)
```

```
Enter the value of x: 10
Enter the value of y: 20
Addition of x and y 30
Subtraction of y from x -10
Multiplication of x and y 200
Division of x by y 0.5
x raised to the power of y 1000000000000000000
The remainder when x is divided by y 10
The floor division of x by y. 0
```

7. Define a variable `value` and assign any numerical value to it. Ask the user to input a new value. Update the variable `value` with the new input and print the updated value.

- **Expected Input:** A numerical value (e.g., "42")
- **Expected Output:** The updated value of the variable.

```
In [16]: value = 10
new_value = input("Enter a value: ") # Take user input
new_value = int(new_value) # Convert input to an integer
new_value = float(new_value)
print(new_value) # Print the updated value
```

Enter a value: 20
20.0

#DAY 3

BODMAS

The BODMAS rule stands for:

1. B - Brackets first
2. O - Orders (exponents and roots, like square roots) next
3. DM - Division and Multiplication, from left to right
4. AS - Addition and Subtraction, from left to right

```
In [17]: result = 10 + 5 * (2 ** 3) - 6 / 2
print(result)
```

47.0

first(2*3)=8 10+58-6/2 second 5*8 = 40 10+40-6/2 third 6/2 = 3 10+40 - 3 fourth 50-3=47

```
In [19]: # 1
result = 10 + 3 * 2 - 8 / 4
#steps
#1. 3*2=6
#10+6-8/4
#2. 8/4=2
#10+6-2
#3. 10+6=16
#16-2
#4. 14 is ans
print(result)
```

14.0

```
In [21]: # 2
result = 4 ** 2 + 5 / 2 * 3
#steps
#1. 4**2 =16
#16+5/2*3
#5/2=2.5
#16+2.5*3
#2. 5*3=7.5
#23
print(result)
```

23.5

```
In [23]: # 3
result = (8 + 4) * 3 / 2
#1. 8+4=12
#2. 12*3=36
#3. 36/2
print(result)
```

18.0

```
In [24]: # 4
result = 16 / 4 + 2 ** 3 - 6
#1.2**3=8
#16/4+8-6
#2.16/4=4
#4+8-6
#12-6
print(result)
```

6.0

```
In [25]: # 5
result = 10 - 3 * (4 + 2) / 5
#1.4+2=6
#10-3*6/5
#10-18/5
#18/5=3.6
#10-3.6
print(result)
```

6.4

```
In [26]: #DAY4
```

```
In [27]: a = "Hi" + 10
print(a)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[27], line 1
----> 1 a = "Hi" + 10
      2 print(a)

TypeError: can only concatenate str (not "int") to str
```

```
In [28]: a = "Puttaparthi" + " " + "Arrow"
print(a)
```

Puttaparthi Arrow

1. Concatenate two strings `str1` and `str2` , and print the result.

- **Expected Input:** `str1 = "Hello", str2 = "World"`
- **Expected Output:** `"HelloWorld"`

```
In [29]: str1="Hello"
str2="World"
print(str1+str2)
```

HelloWorld

2. Ask the user to enter their name and a greeting. Concatenate the name and greeting to form a personalized message and print it.

- **Expected Input:** `name = "John", greeting = "Hi"`
- **Expected Output:** `"Hi John"`

```
In [35]: user=input("Enter a Name: ")
hey=("Hai")
print(hey + " " +user)
```

Enter a Name: siri
Hai siri

3. Create a string `word` and repeat it 5 times. Print the result.

- **Expected Input:** word = "Python"
- **Expected Output:** "PythonPythonPythonPythonPython"

```
In [38]: word=("Python")
print(word*5)
```

PythonPythonPythonPythonPython

4. Ask the user to enter a word and a number. Repeat the word as many times as the given number and print the result.

- **Expected Input:** word = "Hello", number = 3
- **Expected Output:** "HelloHelloHello"

```
In [39]: word="Hello"
num=int(input("Enter a number: "))
res=word*num
print(res)
```

Enter a number: 3
HelloHelloHello

5. Create a string `sentence` and find its length. Print the length of the sentence.

- **Expected Input:** sentence = "This is a sample sentence."
- **Expected Output:** 27

```
In [40]: sentence = "This is a sample sentence."
print(len(sentence))
```

26

6. Ask the user to input a sentence. Find the length of the sentence, and print the last character of the sentence.

- **Expected Output:** Length of the sentence and the last character.

```
In [43]: user=input("Enter a sentence: ")
length=len(user)
char=user[-1]
print(length)
print(char)
```

Enter a sentence: python is amazing
17
g

7. Create two strings `str1` and `str2`. Find the lengths of both strings and concatenate them. Print the concatenated string.

- **Expected Input:** str1 = "Hello", str2 = "World"
- **Expected Output:** "HelloWorld"

```
In [45]: user1=input("Enter a string1 : ")
user2=input("Enter a string2 : ")
length1=len(user1)
length2=len(user2)
print(length1+length2)
print(user1+user2)
```

```
Enter a string1 : Hello
Enter a string2 : World
10
HelloWorld
```

8. Ask the user to enter two words, word1 and word2 . Concatenate the two words with a space in between and print the result.

- **Expected Input:** word1 = "Hello", word2 = "Python"
- **Expected Output:** "Hello Python"

```
In [46]: word1=input("Enter a word1: ")
word2=input("Enter a word2: ")
res=word1 + " " + word2
print(res)
```

```
Enter a word1: Hello
Enter a word2: Python
Hello Python
```

Create a string pattern containing "*" character and repeat it to form a pattern. The pattern should have 5 rows. Print the resulting pattern. Expected Output: * **

```
*****
****
***
**
*
```

```
In [48]: for i in range(1,6):
print("*"*i)
```

```
*
**
***
****
*****
```

```
In [49]: for i in range(2,6):
print("*"*i)
```

```
**
***
****
*****
```

```
In [50]: #DAYS
#TYPE CONVERSION
num_string = "123"
print(type(num_string))
num_int = int(num_string)
print(type(num_int))
```

```
<class 'str'>
<class 'int'>
```

Question 1: Convert the integer 42 to a string.

Expected Input:

value = 42

Expected Output:

result = "42"

```
In [58]: int_value=42
res=str(int_value)
print(type(res))
print(res)
```

```
<class 'str'>
42
```

Question 2: Convert the string "123" to an integer.

Expected Input:

value = "123"

Expected Output:

result = 123

```
In [65]: # Given string value
value = "123"
result = int(value)
print(result) # Output: 123
##it will be get executed in online compiler
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[65], line 3
      1 # Given string value
      2 value = "123"
----> 3 result = int(value)
      4 print(result)
```

TypeError: 'int' object is not callable

Convert the float 3.14 to an integer.

Expected Input:

value = 3.14

Expected Output:

result = 3

```
In [68]: float="3.14"
res= int(float)
print(res)
##it will be get executed in online compiler
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[68], line 2
      1 float=3.14
----> 2 res= int(float)
      3 print(res)
```

TypeError: 'int' object is not callable

```
In [69]: # Given float value
value = 3.14

# Convert float to integer
result = int(value)

# Print the result
print(result)
#it will be get executed in online compiler
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[69], line 5
      2 value = 3.14
      4 # Convert float to integer
----> 5 result = int(value)
      7 # Print the result
      8 print(result)
```

TypeError: 'int' object is not callable

Question 4: Convert the string "5.5" to a floating-point number.

Expected Input:

value = "5.5"

Expected Output:

result = 5.5

```
In [70]: str="5.5"
res=float(str)
print(res)
#it will be get executed in online compiler
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[70], line 2
      1 str="5.5"
----> 2 res=float(str)
      3 print(res)
```

TypeError: 'float' object is not callable

Question 5: Convert the integer 100 to a boolean.

Expected Input:

value = 100

Expected Output:

result = True

```
In [72]: integer=100
res=bool(integer)
print(res)
```

True

Question 6: Convert the boolean True to an integer.

Expected Input:

value = True

Expected Output:

result = 1

```
In [73]: # Given boolean value
value = True
result = int(value)
print(result)
#executed in online complier
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[73], line 3
      1 # Given boolean value
      2 value = True
----> 3 result = int(value)
      4 print(result)
```

TypeError: 'int' object is not callable

Question 7: Convert the string "False" to a boolean.

Expected Input:

value = "False"

Expected Output:

result = True

```
In [74]: value="False"
res=bool(value)
print(res)
```

True

Example: Get the first character of the sentence. Input: "The sun is shining." Output: "T"

```
In [1]: input="The sun is shining"
string=input[0]
print(string)
```

T

Example: Get the last character of the sentence. Input: "She sells seashells by the seashore." Output: "."

```
In [7]: input="She sells seashells by the seashore."
string=input[-1]
print(string)
```

.

Example: Get the character at index 3. Input: "I love Python!" Output: "o"

```
In [1]: Input="I love Python!"
str=Input[3]
print(str)
```

o

Example: Get the second last character of the sentence. Input: "Life is beautiful." Output: "l"

```
In [3]: input="Life is beautiful."  
str=input[-2]  
print(str)
```

l

Example: Get a substring from index 7 to index 14 (exclusive). Input: "Welcome to Python programming." Output: " to Pyt"

```
In [5]: a="Welcome to Python programming."  
str=a[11:14]  
print(str)
```

Pyt

Example: Get a substring from index -9 to -3. Input: "The future is bright." Output: "s brig"

```
In [11]: a="The future is bright."  
str=a[12:18]  
print(str)
```

s brig

Example: Get the first six characters of the sentence. Input: "Good things take time." Output: "Good t"

```
In [3]: a="Good things take time"  
str=a[0:6]  
print(str)
```

Good t

Example: Reverse the sentence using slicing. Input: "Python is awesome!" Output: "!emosewa si nohtyP"

```
In [6]: a="Python is awesome!"  
str=a[::-1]  
print(str)
```

!emosewa si nohtyP

Example: Get the length of the sentence using indexing. Input: "Coding is fun!" Output: 14

```
In [16]: a="Coding is fun!"  
str=len(a)  
print(str)
```

14

#DAY7 Example: Greater than operator (>) Input: 5 > 3 Output: True

```
In [1]: a=5>3  
print(a)
```

True

Example: Less than operator (<) Input: 10 < 20 Output: True

```
In [2]: a=10<20  
print(a)
```

True

Example: Greater than or equal to operator (>=) Input: 7 >= 7 Output: True

```
In [3]: a=7>=7  
print(a)
```

True

Example: Less than or equal to operator (<=) Input: 15 <= 12 Output: False

```
In [4]: a=15<=12  
print(a)
```

False

Example: Equal to operator (==) Input: "hello" == "hello" Output: True

```
In [5]: a="hello"=="hello"  
print(a)
```

True

Example: Not equal to operator (!=) Input: 10 != 20 Output: True

```
In [6]: a=10!=20  
print(a)
```

True

Example: Comparing integers and floats Input: 5.0 > 4 Output: True

```
In [7]: a=5.0>4  
print(a)
```

True

Example: Comparing strings with different cases Input: "Hello" == "hello" Output: False

```
In [8]: a="Hello"=="hello"  
print(a)
```

False

Example: Using relational operators with booleans Input: True == False Output: False

```
In [9]: a=True == False  
print(a)
```

False

Example: Comparing strings Input: "apple" < "banana" Output: True

```
In [10]: a= "apple" < "banana"
print(a)
```

True

Example: Comparing None with a string Input: None == "Python" Output: False

```
In [11]: a=None == "Python"
print(a)
```

False

Example: Mixing different types in comparisons Input: 5 > "3" Output: TypeError

```
In [12]: a=5>"3"
print(a)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[12], line 1
----> 1 a=5>"3"
      2 print(a)
```

TypeError: '>' not supported between instances of 'int' and 'str'

Example: Using relational operators with negative numbers Input: -10 < -5 Output: True

```
In [13]: a=-10<-5
print(a)
```

True

Example: Comparing a string with a number Input: "42" == 42 Output: False

```
In [14]: a="42"==42
print(a)
```

False

Example: Using relational operators with floating-point precision Input: 0.1 + 0.1 + 0.1 == 0.3 Output: False

```
In [15]: a=0.1+0.1+0.1==0.3#becz if we do addition we get a value like 0.30000000...4 so python will
print(a)
```

False

#DAY-8 Logical Operators

1. and : Returns True if both operands are True.
2. or : Returns True if at least one of the operands is True.
3. not : Returns the opposite boolean value of the operand.

True and True ⇒ True True and False ⇒ False False and True ⇒ False False and False ⇒ False

True or True ⇒ True True or False ⇒ True False or True ⇒ True False or False ⇒ False

Example: "and" operator with two True conditions Input: (10 > 5) and ("apple" == "apple") Output: True

```
In [17]: a="apple"=="apple" and 10>5  
print(a)
```

True

Example: "and" operator with one False condition Input: (3 < 2) and ("banana" == "orange") Output: False

```
In [18]: a=3<2 and "banana"=="orange"  
print(a)
```

False

Example: "and" operator with one True and one False condition Input: (5 >= 3) and (10 != 10) Output: False

```
In [19]: a=5>=3 and 10!=10  
print(a)
```

False

Example: "or" operator with two True conditions Input: ("car" == "car") or (7 < 9) Output: True

```
In [20]: a="car"=="car" or 7<9  
print(a)
```

True

Example: "or" operator with one False condition Input: ("dog" == "cat") or (6 < 10) Output: True

```
In [21]: a="dog"=="cat" or 6<10  
print(a)
```

True

Example: "or" operator with both False conditions Input: (2 == 3) or (8 > 15) Output: False

```
In [22]: a=2==3 or 8>15  
print(a)
```

False

Example: "not" operator with True condition Input: not (4 <= 3) Output: True

```
In [23]: a=not 4<=3  
print(a)
```

True

Example: "not" operator with False condition Input: not ("orange" == "orange") Output: False

```
In [25]: a=not "orange"=="orange"  
print(a)
```

False

Example: "not" operator with "and" and "or" Input: not ((5 > 3) and ("apple" != "banana")) Output: False

```
In [26]: a= not 5>3 and "apple"!="banana"#false and true --> true  
print(a)
```

False

Example: "and" and "not" operators combined Input: (10 > 5) and not (3 < 2) Output: True

```
In [28]: a=10>5 and not 3<2  
print(a)
```

True

Example: "or" and "not" operators combined Input: ("cat" == "cat") or not (6 > 10) Output: True

```
In [29]: a="cat"=="cat" or not 6>10  
print(a)
```

True

Example: Using parentheses for grouping expressions Input: ((5 >= 3) and (10 != 10)) or (8 > 15) Output: False

```
In [30]: a=5>=3 and 10!=10 or 8>15  
print(a)
```

False

Example: Combining multiple "and" operators Input: (2 < 5) and (10 == 10) and ("hello" != "world") Output: True

```
In [31]: a=2<5 and 10==10 and "hello"!="world"  
print(a)
```

True

Example: Combining multiple "or" operators Input: (7 < 3) or (5 >= 5) or ("apple" == "apple") Output: True

```
In [32]: a=7<3 or 5>=5 or "apple"=="apple"  
print(a)
```

True

Example: Using "not" operator with an expression Input: not (10 > 5 and "car" != "car") Output: True

```
In [34]: a=not (0>5 and "car"!="car")  
print(a)
```

True

Example: Using "not" operator with "or" and "and" Input: not (5 > 3 or "dog" == "cat" and 7 < 5) Output: False

```
In [35]: a=not 5>3 or "dog"=="cat" and 7<5  
print(a)
```

False

Example: Combining "and" and "or" operators Input: (5 > 3 and "apple" != "banana") or (8 == 8 or 6 < 10)
Output: True


```
In [36]: a=5>3 and "apple"!="banana" or 8==8 or 6<10
print(a)
```

True

Example: Combining "or" and "not" operators Input: ("apple" == "banana" or not (6 > 10)) Output: True

```
In [37]: a="apple"=="banana" or not 6>10
print(a)
```

True

Example: Complex combination of "and", "or", and "not" Input: not (2 < 5 and (7 > 3 or "hello" == "world")) Output: False

```
In [39]: a=not 2<5 and 7>3 or "hello"=="world"
print(a)
```

False

Example: Nested use of "and", "or", and "not" operators Input: (not (5 > 3) and (10 != 10 or "car" == "car")) Output: False

```
In [40]: a=not 5>3 and 10!=10 or "car"=="car"
print(a)
```

True

#DAY-9 Conditional Statements Manam konni conditions use chesi python program flow ni control chestham mowa. Mukhyanga if, elif (short for "else if"), and else. Ee statements manaku condition True or False daani batti pain chesthay! Syntax itluntadi conditional statements di if condition1: # Code block executed if condition1 is True elif condition2: # Code block executed if condition1 is False and condition2 is True else: # Code block executed if both condition1 and condition2 are False

```
In [1]: # Example 1: Using 'if' to check a condition
age = 18

if age >= 18:
    print("You are an adult.")
```

You are an adult.

```
In [2]: # Example 2: Using 'if' and 'else' to check a condition
age = 15

if age >= 18:
    print("You are an adult.")
else:
    print("You are a minor.")
```

You are a minor.

```
In [3]: # Example 3: Using 'if', 'elif', and 'else' to check multiple conditions
age = 25

if age < 18:
    print("You are a minor.")
elif age >= 18 and age < 65:
    print("You are an adult.")
else:
    print("You are a senior citizen.")
```

You are an adult.

Question 1: Write a program that takes a number as input and prints "Even" if it's an even number, and "Odd" if it's an odd number. Expected Input: Enter a number: 7

Expected Output: Odd

```
In [7]: a=int(input("Enter a num: "))
if (a%2==0):
    print("even")
else:
    print("odd")
```

Enter a num: 7
odd

Question 2: Write a program that takes two numbers as input and prints the larger number. Expected Input: Enter the first number: 15 Enter the second number: 22

Expected Output: 22

```
In [10]: a=input("Enter num1: ")
b=input("Enter num2: ")
if a>b:
    print("15")
else:
    print("22")
```

Enter num1: 15
Enter num2: 22
22

Question 3: Write a program that takes a character as input and prints "Vowel" if it's a vowel (a, e, i, o, u), and "Consonant" otherwise. Expected Input: Enter a character: a

Expected Output: Vowel

```
In [13]: a=input("enter a charcater: ")
if a in 'aeiou':
    print("Vowel")
else:
    print("constant")
```

enter a charcater: a
Vowel

Question 4: Write a program that takes a year as input and prints "Leap Year" if it's a leap year, and "Not a Leap Year" otherwise. Expected Input: Enter a year: 2024

Expected Output: Leap Year

```
In [16]: year=int(input("Enter a year: "))
if (year % 4 == 0):#if take a 2004%4 ==0 so its a Leap its divisible by 4 so its a Leap yea
    print("leap")
else:
    print("not a leap")
```

Enter a year: 2004
leap

Question 5: Write a program that takes a grade as input (A, B, C, D, or F) and prints "Pass" if it's A, B, C, or D, and "Fail" if it's F. Expected Input: Enter your grade: C

Expected Output: Pass

```
In [25]: grade=input("A,B,C,D or F: ")
if grade in ('A,B,C,D'):
    print("Pass")
else:
    print("Fail")
```

A,B,C,D or F: C
Pass

In []: