**1.Description:**

You will design and implement a program that prompts the user for an input with:

How can I help you?

User input should be one of three possible kinds:

Please convert \*\*\*

Please convert \*\*\* using \*\*\*

Please convert \*\*\* minimally

If the user input is not of this form, with any occurrence of \*\*\* an arbitrary nonempty sequence of non-space symbols, then the program should print out:

I don't get what you want, sorry mate!

and stop.

**2.First Kind of Input**

In case the user inputs Please convert \*\*\*, then \*\*\* should be either a strictly positive integer (whose representation should not start with 0) that can be converted to a Roman number (hence be at most equal to 3999), or a valid Roman number; otherwise, the program should print out:

Hey, ask me something that's not impossible to do!

and stop.

If the input is as expected, then the program should perform the conversion, from Arabic to Roman or from Roman to Arabic, and print out the result in the form:

Sure! It is \*\*\*

**3. Second Kind of Input**

In case the user inputs Please convert \*\*\* using \*\*\*, then the first \*\*\* should be a strictly positive integer (whose representation should not start with 0) or a sequence of (lowercase or uppercase) letters and the second \*\*\* should be a sequence of distinct (lowercase or uppercase) letters.

Moreover:

• the second \*\*\* is intended to represent a sequence of so-called generalised Roman symbols. The classical Roman symbols corresponding to the sequence MDCLXVI, whose rightmost element is meant to represent 1, the second rightmost element 5, the third rightmost element 10, etc.

• if it is not an integer, the first \*\*\* is intended to represent a so-called generalised Roman number, that is, a sequence of generalised Roman symbols that can be decoded using the provided sequence of generalised Roman symbols similarly to the way Roman numbers are represented.

If that is not the case, or if it is not possible to convert the first \*\*\* from Arabic to generalised Roman or from generalised Roman to Arabic, then the program should print out:

Hey, ask me something that's not impossible to do!

and stop.

If the input is as expected and the conversion can be performed, then the program should indeed perform the conversion, from Arabic to generalised Roman or from generalised Roman to Arabic, and print out the result in the form:

Sure! It is \*\*\*

**4.Third Kind of Input**

In case the user inputs Please convert \*\*\* minimally, then \*\*\* should be a sequence of (lowercase or uppercase) letters. The program will try and view \*\*\* as a generalised Roman number with respect to some sequence of generalised Roman symbols. If that is not possible, then the program should print out:

Hey, ask me something that's not impossible to do!

and stop.

Otherwise, the program should find the smallest integer that could be converted from \*\*\*, viewed as some generalised Roman number, to Arabic, and output a message of the form

Sure! It is \*\*\* using \*\*\*

**5. Sample Outputs (or Test Cases)**

Here are a few tests together with the expected outputs. The outputs of your program should be exactly as shown:

$ python3 roman\_arabic.py

How can I help you? Please do my assignment...

I don't get what you want, sorry mate!

$ python3 roman\_arabic.py

How can I help you? please convert 35

I don't get what you want, sorry mate!

$ python3 roman\_arabic.py

How can I help you? Please convert 035

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you? Please convert 4000

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you? Please convert IIII

Hey, ask me something that's not impossible to do!

$python3 roman\_arabic.py

How can I help you? Please convert IXI

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you? Please convert 35

Sure! It is XXXV

$ python3 roman\_arabic.py

How can I help you? Please convert 1982

Sure! It is MCMLXXXII

$ python3 roman\_arabic.py

How can I help you? Please convert 3007

Sure! It is MMMVII

$ python3 roman\_arabic.py

How can I help you? Please convert MCMLXXXII

Sure! It is 1982

$ python3 roman\_arabic.py

How can I help you? Please convert MMMVII

Sure! It is 3007

$ python3 roman\_arabic.py

How can I help you? Please convert 123 by using ABC

I don't get what you want, sorry mate!

$ python3 roman\_arabic.py

How can I help you? Please convert 123 ussing ABC

I don't get what you want, sorry mate!

$ python3 roman\_arabic.py

How can I help you? Please convert XXXVI using VI

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you? Please convert XXXVI using IVX

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you? Please convert XXXVI using XWVI

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you? Please convert I using II

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you?

Please convert \_ using \_

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you?

Please convert XXXVI using XVI

Sure! It is 36

$ python3 roman\_arabic.py

How can I help you?

Please convert XXXVI using XABVI

Sure! It is 306

$ python3 roman\_arabic.py

How can I help you?

Please convert EeDEBBBaA using fFeEdDcCbBaA

Sure! It is 49036

$ python3 roman\_arabic.py

How can I help you?

Please convert 49036 using fFeEdDcCbBaA

Sure! It is EeDEBBBaA

$ python3 roman\_arabic.py

How can I help you?

Please convert 899999999999 using AaBbCcDdEeFfGgHhIiJjKkLl

Sure! It is Aaaabacbdcedfegfhgihjikjlk

$ python3 roman\_arabic.py

How can I help you?

Please convert ABCDEFGHIJKLMNOPQRST using AbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStT

Sure! It is 11111111111111111111

$ python3 roman\_arabic.py

How can I help you?

Please convert 1900604 using LAQMPVXYZIRSGN

Sure! It is AMAZING

$ python3 roman\_arabic.py

How can I help you?

Please convert ABCD minimally using ABCDE

I don't get what you want, sorry mate!

$ python3 roman\_arabic.py

How can I help you?

Please convert ABCD minimaly

I don't get what you want, sorry mate!

$ python3 roman\_arabic.py

How can I help you?

Please convert 0I minimally

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you?

Please convert ABAA minimally

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you?

Please convert ABCDEFA minimally

Hey, ask me something that's not impossible to do!

$ python3 roman\_arabic.py

How can I help you?

Please convert MDCCLXXXVII minimally

Sure! It is 1787 using MDCLXVI

$ python3 roman\_arabic.py

How can I help you?

Please convert MDCCLXXXIX minimally

Sure! It is 1789 using MDCLX\_I

$ python3 roman\_arabic.py

How can I help you?

Please convert MMMVII minimally

Sure! It is 37 using MVI

$ python3 roman\_arabic.py

How can I help you?

Please convert VI minimally

Sure! It is 4 using IV

$ python3 roman\_arabic.py

How can I help you?

Please convert ABCADDEFGF minimally

Sure! It is 49269 using BA\_C\_DEF\_G

$ python3 roman\_arabic.py

How can I help you?

Please convert ABCCDED minimally

Sure! It is 1719 using ABC\_D\_E

**6.Hints**

6.1Explaining the following example of the third kind of input

(Please convert \*\*\* minimally):

$ python3 roman\_arabic.py

How can I help you?

Please convert ABCADDEFGF minimally

Sure! It is 49269 using BA\_C\_DEF\_G

First, remember the two important Roman numeral rules below:

1. A Roman symbol is repeated three times but not more than that. However, the symbols V (5), L (50) and D (500) are never repeated.

2. The Roman symbols V (5), L (50) and D (500) are never written to the left of a symbol of greater value, i.e., V (5), L (50) and D (500) are never subtracted. The symbol I (1) can be subtracted from V (5) and X (10) only. The symbol X can be subtracted from L (50) and C (100) only.

Note also that "minimally" means we are looking for a generalised Roman symbols that can convert the given numeral into a smallest integer number.

Let us start assigning Roman numeral values from the right-hand side such that the value is minimum.

Starting with F, we can see it is repeated and we have to assign the minimum value to FGF in order to assign the minimum value to F. From a number of various combinations, we know that the only possible solution here is F=10 and G=1 (try out combinations of 1, 5, 10 here to see why this is the right one). Thus FGF=19.

Let us move now to the next element, which is E. We also need to consider the element after E in order to assign a smaller combination, if possible, in this case. The next element is D, which is repeated and therefore cannot be less than E. Thus, we assign E the smallest number not used yet, which is 50. Moving on to D, since it is repeated, it cannot be greater than the next element A. Thus, we assign the smallest number not yet used which is 100 to D.

Till now, our number DDEFGF is resulting in 269 using DEF\_G (value 5 not assigned).

The next element is A and it is repeated. To assign a value to A, we must assign a value so that ABCA does not violate Roman numeral rules. That is, A < B and B > C. Because of AB (A and B being next to each other), we cannot assign A as 500 (500 cannot be subtracted from any number)

Let us say we assign 1000 to A. Then B can be either 5000 or 10000. B cannot be 5000 because that would mean C can only be 500. Also, B cannot be 10000 as it would mean C should be 5000 or 500 (both are invalid assignments).

Let us try to assign 10000 to A (it cannot be assigned 5000 since it is repeated). B can be either 50000 or 100000. If B is 50000, C can be either 5000, 1000 or 500. C cannot be 5000 or 500 (since they be subtracted from any number). C can be 1000.

Consequently, the smallest we can come up with here is 10000 for A, 50000 for B, and 1000 for C, and ABCA = 50000 - 10000 + 10000 - 1000 = 49000.

Thus, the total becomes 49269 using BA\_C\_DEF\_G (values 5, 500 and 5000 not assigned).

**7.More examples about the third kind of input**

(Please convert \*\*\* minimally):

$ python3 roman\_arabic.py

How can I help you? Please convert AZERTY minimally

Sure! It is 444 using ZAREYT

$ python3 roman\_arabic.py

How can I help you? Please convert XXXVVVIII minimally

Sure! It is 333 using X\_V\_I

$ python3 roman\_arabic.py

How can I help you? Please convert AhZhJ minimally

Sure! It is 691 using Ah\_Z\_J

$ python3 roman\_arabic.py

How can I help you? Please convert BCBC minimally

Hey, ask me something that's not impossible to do!