|  |
| --- |
| So far, we have only heard of Python's powers. Now, we will witness（证明） them! |
| Powers （幂次方）or exponents （范例）in Python can be calculated using the built-in（内置的） power function. |
| This is very helpful in computations where you have to print the resultant （结果的、合成的）% mod. |
| Triangle （三角）Quest |
| in order of their occurrence （发生、事件）from top to bottom |
| Parsing  （解析）is the process of syntactic （语法）analysis of a string of symbols（符号、象征）. It involves resolving a string into its component（组成） parts and describing their syntactic roles. 解析是一串符号的句法分析过程。 它涉及将字符串解析为其组成部分并描述其语法角色。 |
| This method is called when a comment（描述、评论） is encountered（遇见） (e.g. <!--comment-->).  The data argument is the content inside the comment tag: 数据参数是注释标签内的内容 |
| Symmetric difference (对称差分) |
| Distinct ()明显的、独特的、有区别的 |
| The students of District College have subscriptions （订阅）to English and French newspapers. |
| Stuart has to make words starting with consonants.(辅音) |
| Kevin has to make words starting with vowels. （元音） |
| You are given a polynomial (多项式)P of a single indeterminate（不确定的、模糊的）,x. |
| Your task is to print a palindromic (回文三角)triangle of size |
| This tool returns length subsequences of elements （长度子序列）from the input iterable （输入迭代）allowing individual elements to be repeated more than once.（允许单个元素重复多次）  Combinations are emitted （排列）in lexicographic （字典）sorted order（排序次序）. So, if the input iterable is sorted, the combination tuples will be produced in sorted order. 此工具返回来自输入迭代的元素的长度子序列，从而允许单个元素重复多次。组合按字典顺序排列。 所以，如果输入迭代被排序，组合元组将按排序顺序生成。 |
| When stacking (堆叠)the cubes, you can only pick up either the leftmost or the rightmost cube each time. Print "Yes" if it is possible to stack the cubes. Otherwise, print "No". Do not print the quotation marks. |
| There is a horizontal row of cubes. The length of each cube is given. You need to create a new vertical （垂直的）pile of cubes. The new pile should follow these directions: 有一排水平方块。给出每个立方体的长度。 您需要创建一个新的垂直堆立方体。 新的堆应该遵循这些方向： |
| The itertools module standardizes a core set of fast, memory efficient tools that are useful by themselves or in combination. Together, they form an iterator algebra making it possible to construct specialized tools succinctly and efficiently in pure Python.  itertools模块标准化了一组核心快速，高效的内存工具，这些工具本身或组合都很有用。 它们一起形成了一个迭代器代数，可以在纯Python中简洁高效地构建专用工具。 |
| you can select any indices（指数、索引） |
| The linalg.det tool computes the determinant （行列式）of an array |
| The linalg.eig computes the eigenvalues （特征值）and right eigenvectors （右特征向量）of a square array.  linalg.eig计算方阵的特征值和右特征向量。 |
| The linalg.inv tool computes the (multiplicative（乘法）) inverse （逆）of a matrix. |
| Your task is to find the determinant.（行列式，决定因素） |
| 3-dimensional Cartesian coordinate system 三维笛卡尔坐标 |
| not radian 没有弧度  means the dot product of x与y的点积 |
| 意味着向量的叉积means the cross product of vectors |
| The expression re.findall() returns all the non-overlapping matches of patterns in a string as a list of strings. 表达式re.findall（）返回字符串中模式的所有非重叠匹配作为字符串列表。 |
| The expression re.finditer() returns an iterator yielding MatchObject instances over all non-overlapping matches for the re pattern in the string. 表达式re.finditer（）返回一个迭代器，它在字符串中的re模式的所有非重叠匹配上产生MatchObject实例。 |
| These expressions return the indices of the start and end of the substring matched by the group. 这些表达式返回组匹配的子串的开始和结束的索引。 |
| The polyint tool returns an antiderivative 反导数(indefinite integral 不定积分) of a polynomia |
| The polyder tool returns the derivative of the specified order of a polynomial。返回多项式的指定顺序的导数 |
| The polyval tool evaluates the polynomial at specific value. 特定值评估多项式 |
| The polyfit tool fits a polynomial of a specified order to a set of data using a least-squares approach. polyfit工具使用最小二乘法将指定顺序的多项式拟合到一组数据中。 |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |