

Question 1

If we list all the natural numbers below **10** that are multiples of **3** or **5**, we get **3, 5, 6** and **9**. The **sum** of these multiples is **23**.

Find the **sum** of all the multiples of **3** or **5** below **1000**.

Question 2

Palindrome Permutation: Given a string, write a function to check if it is a permutation of a palindrome. A palindrome is a word or phrase that is the same forwards and backwards. A permutation is a rearrangement of letters. The palindrome does not need to be limited to just dictionary words.

EXAMPLE:

Input: Tact Coa

Output: True (permutations: "taco cat", "atco cta", etc.)

Question 3

Implement an algorithm to determine if a string has all unique characters. You cannot use additional data structures (e.g. HashMap). Recommended time complexity: $O(n)$ or lesser

Example:

Input: 'Banana'

Output: False (as there are 2 'n' and 3 'a')

Question 4

A child is running up a staircase and can hop either 1 step, 2 steps, or 3 steps at a time. Implement a method to count how many possible ways the child can run up the stairs and stop when the child is n steps or more up the stairs.

Question 5

Implement the 'reshape' function of numpy.

Given an array $A = (1, 2, 3, \dots, L)$, this is a 1D array.

Implement the reshape function to reshape this array into multi-dimensional array. Write a program as follows,

```
threeD_array = reshape3D(A, L1, L2, L3) # where we suppose  $L1 * L2 * L3 = L$ 
fiveD_array = reshape5D(A, K1, K2, K3, K4, K5)
```

Also implement the functions `get()`, `index()` and `coordinates()` such as:

- `threeD_array.get(x,y,z)` will get the value at x,y,z coordinate location
- `threeD_array.index(x,y,z)` will give the corresponding index in of this element in A
- `threeD_array.coordinates(index)` will give the x,y,z coordinate in the 3D array
- `fiveD_array.get(u,w,x,y,z)` will get the value at u,w,x,y,z coordinate location
- `fiveD_array.index(u,w,x,y,z)` will give the corresponding index in of this element in A
- `fiveD_array.coordinates(index)` will give the u,w,x,y,z coordinate in the 3D array

then

use your code to show that for any **index**, 1 to L,

```
[x,y,z] = threeD_array.coordinates(index)
```

```
v3 = threeD_array.get(x,y,z)
```

```
idx3 = threeD_array.index(x,y,z)
```

show that idx3 = index

```
[u5,w5,x5,y5,z5] = fiveD_array.coordinates(idx3)
```

```
v5 = fiveD_array.get(u5,w5,x5,y5,z5)
```

show that v3 = v5

run your code for $L = 10,000$,

$L1 = 5, L2 = 10, L3 = 10,000/(5*10)$

$K1=3, K2=4, K3=2, K4=20, K5=10,000/(5*4*2*10)$

Question 6

Given a one dimensional array of boxes of length n . Each box to be filled with either one red bead or one blue bead and all boxes are to be filled. Once the boxes are filled, there are $n-1$ neighboring pairs of beads. Calculate a score this way, if two neighboring beads are of the same color, score = -1, otherwise score = +1. The total score is the sum of scores of all neighboring beads.

e.g. given this array with red (r) and blue (b) beads : r b r r r b r b, total score = $+1+1-1-1+1+1+1$

- (a) What are the minimum and maximum possible scores among all possible ways of filling the box?
- (b) Derive a general formula for all possible scores for an array of length n
- (c) Given $n = 29$, what are all possible scores. For each possible score, how many ways can we fill the array to achieve this score?
- (d) Given $n = 938103$, what are all possible scores and how many ways can we fill the array to achieve this score?

- (e) Given $n = 51$, number of red beads is 17 and number of blue beads is $51 - 17 = 34$. How many possible ways can you fill the array of boxes? For each way, what is the score of the array?
- (f) Given $n = 28,173,831$, the number of red beads is 12,249,491 and the remaining boxes are filled with blue beads. How many possible ways can you fill the array of boxes?