**Functions**

1. Given two DNA strands. Find the hamming distance between them. (*The Hamming distance is only defined for sequences of equal length*)

|  |  |
| --- | --- |
| **Input** | **Output** |
| ‘GAGCCTACTAACGGGAT’ ‘CATCGTAATGACGGCCT’ | 7 |

1. Given a DNA strand, return its RNA complement (per RNA transcription).  
   Both DNA and RNA strands are a sequence of nucleotides.  
     
   The four nucleotides found in DNA are adenine (A), cytosine (C), guanine (G) and thymine (T).  
   The four nucleotides found in RNA are adenine (A), cytosine (C), guanine (G) and uracil (U).  
   Given a DNA strand, its transcribed RNA strand is formed by replacing each nucleotide with its complement:  
     
   G -> C  
   C -> G  
   T -> A  
   A -> U

|  |  |
| --- | --- |
| **Input** | **Output** |
| GATGCCGAT | CUACGGCUA |
| CGTAATAGGT | GCAUUAUCCA |

1. Compute the prime factors of a given natural number.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 456 | [2, 3, 19] |
| 3465 | [3, 5, 7, 11] |

1. Find the difference between the square of the sum and the sum of the squares of the first **n** natural numbers.  
   (1 + 2 + ... + 10)² = 55² = 3025  
   1² + 2² + ... + 10² = 385  
   3025 - 385 = 2640

1. Write a function to print all strong numbers before the given number. *(Strong numbers are the numbers whose sum of factorial of digits is equal to the original number.)*

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 | “1, 2” |
| 160 | “1, 2, 145” |

1. Given an array of numbers. Write a function to separate odd and even numbers in different arrays.

|  |  |
| --- | --- |
| Input | Output |
| [45, 12, 3, 6, 17, 0] | [45, 3, 17]  [12, 6, 0] |
| [1, 3, 5, 9] | [1, 3, 5, 9]  [] |

1. Write a function that calculates sum, difference, multiplication and division between given array elements depending on passed operation symbol. Write appropriate function for each operation.
2. Given a phone number. Clean up it, so it is valid.  
   The rules are as follows:  
     
   If the phone number is less than 10 digits assume that it is bad number  
   If the phone number is 10 digits assume that it is good  
   If the phone number is 11 digits and the first number is 1, trim the 1 and use the last 10 digits  
   If the phone number is 11 digits and the first number is not 1, then it is a bad number  
   If the phone number is more than 11 digits assume that it is a bad number.  
     
   Ignore spaces.

|  |  |
| --- | --- |
| **Input** | **Output** |
| “455678” | “Bad number” |
| “339 5656888” | “3395656888” |
| “10008989562” | “0008989562” |
| “45231489562” | “Bad number” |
| “1232356897452” | “Bad number” |

1. Given a word and a list of possible anagrams, select the correct sublist.

|  |  |
| --- | --- |
| **Input** | **Output** |
| “listen”, ["enlists" "google" "inlets" "banana"] | [“inlets”] |
| “pencil”, [“licnep”, “circular”, “pupil”, “nilcpe”, “leppnec”] | [“licnep”, ]nilcpe |

1. Write a function, which receives a string, finds possible largest numbers in the string and returns their sum.

|  |  |
| --- | --- |
| **Input** | **Output** |
| “FwrtY45KHL120mn10P” | 175 |
| “Wert12lop-12” | 0 |

1. Write a function which receives two strings and removes appearances of the second string from the first one.

|  |  |
| --- | --- |
| **Input** | **Output** |
| “This is some text.”, “is” | “Th some text.” |
| “Yes, London. You know: fish, chips, cup ‘o tea, bad food, worse weather”, “o” | “Yes, Lndn. Yu knw: fish, chips, cup ‘ tea, bad fd, wrse weather” |

1. Write a function to compute a new string from the given one by moving the first char to come after the next two chars, so "abc" yields "bca". Repeat this process for each subsequent group of 3 chars. Ignore any group of fewer than 3 chars at the end.

|  |  |
| --- | --- |
| Input | Output |
| “dfgjkloyp” | “fgdkljypo” |
| “aweyoolp” | “weaooylp” |