**Refer to a course material and M06 Kahoot questions.**

**Q1.** Creat a function named ‘**CountSDigitNum**’ that counts the number of **single digit numbers** in a given **row vector**, ‘vec’, that contains positive integers.

|  |
| --- |
| **function** out = CountSDigitNum(vec) |

**Q2.** Creat a function named ‘**CountSDigitNumNew**’ that counts the number of **single digit numbers** in a given array ‘in\_array’, which is **either scalar, vector or matrix**. Note that ‘in\_array’ contains positive integers.

|  |
| --- |
| **function** out = CountSDigitNumNew (in\_array) |

**Q3.** Creat a function named ‘**SumEven**’ that **sums all even numbers** in a given array ‘in\_array’, which is either scalar, vector or matrix. ‘in\_array’ contains integer values.

|  |
| --- |
| **function** out = SumEven(in\_array) |

**Q4.** Creat a function named ‘**EvenArr**’ that **sums** and **counts** all even numbers in a given array ‘in\_arr’, which is either scalar, vector or matrix. ‘**c\_in**’: count and ‘**s\_in**’: sum

|  |
| --- |
| **function** [c\_in, s\_in] = EvenArr (in\_arr) |

**Q5.** Creat a function named ‘**PosArr**’ that sums, counts, and find their locations of (linear index) all positive numbers in a given array ‘in\_arr’. ‘**c\_in**’: count, ‘**s\_in**’: sum, and ‘**l\_in**’: liner index

|  |
| --- |
| **function** [c\_in, s\_in, l\_in] = PosArr (in\_arr) |

**Q6.** Creat a function named ‘**PosArrNew’** that sums, counts, **or** find their locations (linear index) of all positive numbers in a given array ‘in\_arr’ **depending on ‘opt’. ‘opt’** takes ‘sum’, ‘count’, and ‘loc’ to select its operation.

|  |
| --- |
| **function** out = PosArrNew(in\_arr, opt) |

**Q7.** Creat a function named ‘**MZ\_RL**’ that move all zeros in each row in ‘in\_arr’ **all the way to the front of the correponding row**. The order of the remaining non-zero elements in each row should be preserved. A resulting matrix is assigned to ‘out’. ‘in\_arr’ is a matrix.

|  |
| --- |
| **function** out = MZ\_RL (in\_arr)  end  **function** row\_new = MoveZeroRLVec(row)  row\_new = zeros(1, numel(row));  lg\_vec = row == 0;  num = sum(lg\_vec);  row\_new(num+1:end) = row(~lg\_vec);  end |