Lab 06
OOP – BSDS

Task 01: Write following functions for a 1D list of numbers:

a) Print list in matrix/ row-column form (Function has 3 parameters list, rows & columns)

It is given that list has at least rows x columns elements. See example:

Consider:

1 2 3 4 5 6 7 8 9 10 11 12

* Print in 3 rows and 4 columns

1 2 3 4

5 6 7 8

9 10 11 12

* Print in 2 rows and 6 columns

1 2 3 4 5 6

7 8 9 10 11 12

b) Convert 1D list into 2D list and return 2D list (Again, parameters are 1D list, rows & columns)

Hint: Take 2D list say of zeros (syntax in next line) and place element of 1D list into 2D list

two_d_list = =[[0 for i in range(columns)] for j in range(rows)]

c) Place 1D list into 2D list at location (i, j), where 2D list has elements greater than equal to 1D list. (There are 6parameters 2D list, 1D list, i, j, rows & columns). See example:

Consider 2D list

11 12 13 14 15

16 17 18 19 20

21 22 23 24 25

26 27 28 29 30

Consider 1D list

1 2 3 4 5 6

→ Place 1D list at position **1, 2** in 2 rows & 3 columns. The 2D list after placement:

11 12 13 14 15

16 17 **1 2 3**

21 22 **4 5 6**

26 27 28 29 30

Task 02: Create following classes with required data members and functions. Write separate main program to test your class:

a) Class student has members: roll no (positive integer value), semester no (positive integer value) & cgpa (positive float value). Write initializer function (with all data members as parameters to initialize data members), one setter function for all members, two individual setter functions for semester no & cgpa. Write two getter functions for semester no & cgpa. Write one display function to show objects like:
`Roll No: 4 Semester No: 3 CGPA: 3.15`

b) Class Batsman has members: player id (positive integer value), no of matches (positive integer value), runs (positive integer value), average (positive float value). Keep data members private. Write initializer function with one parameter (player id), assign 0 to remaining data members. Write one private function, update average (Simply divide total runs by number of matches and assign to average). Write one setter function to add match with one parameter score (add 1 to number of matches, add score to current score and call private function to update average). Write display function to print the batsman in following format:

Player Id: Runs: Average: