Assignment #4

READ THE ENTIRE DOCUMENT – There are two programs to submit. Use the specified name for your scripts

Any work you submit for this assignment should be authored entirely by yourself. Assistance is permitted from the instructor or teaching assistants only. All submitted programming assignments are subject to originality verification through software designed and used for the Measure Of Software Similarity (MOSS).

1) *(multiples.py)* Using nested loops, create a multiplication table. Prompt the user for the range of values to include. Validate their start value to be positive and less than 10000 and their end value to be positive, less than 1000 and greater than the start value. Be sure to label the chart and ensure your formatting matches the sample program run.

Sample run #1:

```
Enter a positive starting value < 1000: 0
0  is not a valid choice, try again
Enter a positive starting value < 1000: 1
Enter a positive ending value less than 1000 and greater than the starting value: 10
```

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Sample run #2:

```
Enter a positive starting value < 1000: 250
Enter a positive ending value less than 1000 and greater than the starting value: 250
250  is not a valid choice, try again
Enter a positive ending value less than 1000 and greater than the starting value: 1000
1000  is not a valid choice, try again
Enter a positive ending value less than 1000 and greater than the starting value: 265
```

	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265
250	62500	62750	63000	63250	63500	63750	64000	64250	64500	64750	65000	65250	65500	65750	66000	66250
251	62750	63001	63252	63503	63754	64005	64256	64507	64758	65009	65260	65511	65762	66013	66264	66515
252	63000	63252	63504	63756	64008	64260	64512	64764	65016	65268	65520	65772	66024	66276	66528	66780
253	63250	63503	63756	64009	64262	64515	64768	65021	65274	65527	65780	66033	66286	66539	66792	67045
254	63500	63754	64008	64262	64516	64770	65024	65278	65532	65786	66040	66294	66548	66802	67056	67310
255	63750	64005	64260	64515	64770	65025	65280	65535	65790	66045	66300	66555	66810	67065	67320	67575
256	64000	64256	64512	64768	65024	65280	65536	65792	66048	66304	66560	66816	67072	67328	67584	67840
257	64250	64507	64764	65021	65278	65535	65792	66049	66306	66563	66820	67077	67334	67591	67848	68105
258	64500	64758	65016	65274	65532	65790	66048	66306	66564	66822	67080	67338	67596	67854	68112	68370
259	64750	65009	65268	65527	65786	66045	66304	66563	66822	67081	67340	67599	67858	68117	68376	68635
260	65000	65260	65520	65780	66040	66300	66560	66820	67080	67340	67600	67860	68120	68380	68640	68900
261	65250	65511	65772	66033	66294	66555	66816	67077	67338	67599	67860	68121	68382	68643	68904	69165
262	65500	65762	66024	66286	66548	66810	67072	67334	67596	67858	68120	68382	68644	68906	69168	69430
263	65750	66013	66276	66539	66802	67065	67328	67591	67854	68117	68380	68643	68906	69169	69432	69695
264	66000	66264	66528	66792	67056	67320	67584	67848	68112	68376	68640	68904	69168	69432	69696	69960
265	66250	66515	66780	67045	67310	67575	67840	68105	68370	68635	68900	69165	69430	69695	69960	70225

2) (*amoritization.py*) In assignment #2, you created a program to calculate the monthly payment for a loan, given the starting amount, annual interest rate (APR) and term (in years).

An amortization table displays the monthly interest paid and resulting principal due over the course of the loan. You will create an amortization table for the values provided by the user. To accomplish this, for each month, take the remaining principal due and multiple by the annual interest rate – this is the interest that would be due over the course of a year. Divide by 12 to find the interest due this month. The payment (calculated in the first step), is first applied to this interest due, and the remaining amount is applied to principal.

You are to write a program that prompts the user for initial loan amount, APR and term, all validated to be positive. The starting amount and APR can be decimal values; the term is a whole number of years. Produce an amortization schedule for all months of the loan. For each month, indicate the month number, the interest paid for the month, the cumulative interest to date and the remaining principal.

In the provided example, the user takes a loan for $25,000 at 5% interest for a term of 4 years. The monthly payment would be $575.73. The interest in the first month would be annual interest divided by 12 months → 25,500 * 5/100 /12 = $104.17. The remaining amount applied to the principal would be $575.73 - $104.17 = $471.56, bringing the principal due down to $24528.44. For subsequent months, the new principal value is used.

Use a while loop to validate all input values to be positive.

Use a for loop to print the table.

```
Original loan amount? 25000
Annual interest rate? 5
Years? 4
Payment is $ 575.73
Month   Interest  Cumulative    Principal
       this month  Interest
   1     104.17     104.17       24528.44
   2     102.20     206.37       24054.91
   3     100.23     306.60       23579.41
   4      98.25     404.84       23101.92
   5      96.26     501.10       22622.45
   6      94.26     595.36       22140.98
   7      92.25     687.62       21657.51
   8      90.24     777.86       21172.02
   9      88.22     866.07       20684.50
  10      86.19     952.26       20194.96
  11      84.15    1036.40       19703.37
  12      82.10    1118.50       19209.74
  13      80.04    1198.54       18714.05
  14      77.98    1276.52       18216.30
  15      75.90    1352.42       17716.47
  16      73.82    1426.24       17214.56
  17      71.73    1497.96       16710.55
  18      69.63    1567.59       16204.45
  19      67.52    1635.11       15696.24
  20      65.40    1700.51       15185.91
  21      63.27    1763.79       14673.46
  22      61.14    1824.93       14158.87
  23      59.00    1883.92       13642.13
  24      56.84    1940.76       13123.24
  25      54.68    1995.44       12602.19
  26      52.51    2047.95       12078.97
  27      50.33    2098.28       11553.57
  28      48.14    2146.42       11025.98
```

29	45.94	2192.36	10496.19
30	43.73	2236.10	9964.20
31	41.52	2277.61	9429.98
32	39.29	2316.91	8893.55
33	37.06	2353.96	8354.87
34	34.81	2388.77	7813.95
35	32.56	2421.33	7270.78
36	30.29	2451.63	6725.35
37	28.02	2479.65	6177.64
38	25.74	2505.39	5627.65
39	23.45	2528.84	5075.37
40	21.15	2549.99	4520.79
41	18.84	2568.82	3963.89
42	16.52	2585.34	3404.68
43	14.19	2599.53	2843.14
44	11.85	2611.37	2279.25
45	9.50	2620.87	1713.02
46	7.14	2628.01	1144.43
47	4.77	2632.77	573.46
48	2.39	2635.16	0.12