Homework: PHP Flow Control

This document defines the homework assignments from the "PHP Basics" Course @ Software University. Please submit as homework a single **`zip`** / **`rar`** / **`7z`** archive holding the solutions (source code) of all below described problems.

Problem 1. Square Root Sum

Write a PHP script **SquareRootSum.php** that displays a **table** in your browser with **2 columns**. The first column should contain a **number** (even numbers from 0 to 100) and the second column should contain **the square root of that number**, rounded to the **second digit after the decimal point**. The **last row** of the table should contain the **sum** of all values in the **Square** column. *Styling the page is optional.* Partial output comes below:

Output

Number	Square
0	0
2	1.41
4	2
6	2.45
8	2.83
...	...
94	9.7
96	9.8
98	9.9
100	10
Total:	338.04

Problem 2. Rich People's Problems

You are a very rich billionaire with an unhidden passion for cars. You like certain car manufacturers but you don't really care about anything else, and that's why you need your own randomizing algorithm that helps you decide how many and what color cars you should buy. Write a PHP script **CarRandomizer.php** that receives **a string of cars** from an **input HTML form**, separated by a comma and space (", "). It then prints **each car**, a random **color** and a random **quantity** in a **table** like the one shown below. Use colors by your choice. Use as quantity a random number in range [1...5]. Styling the page is *optional.* Examples:

Input

```
"Mitsubishi, Maseratti, Maybach"
```

Enter cars Mitsubishi, Maseratti, Maybach Show result

Output

Enter cars Mitsubishi, Maseratti, Maybach Show result

Car	Color	Count
Mitsubishi	yellow	2
Maseratti	green	1
Maybach	black	3

Problem 3. **Show Annual Expenses**

Write a PHP script **AnnualExpenses.php** that receives **n** years from an **input HTML form** and creates a **table** (like the one shown below) with **random expenses** by **months** and the corresponding **years** (n years back). For example, if N is 10, create a table that shows the expenses for each month for the last 10 years. Add a "**Total**" column at the end, showing the total expenses for the same year. The random expenses in the table should be in the range [0…999]. Styling the page is *optional.* Examples:

Input

Enter number of years: 5 Show costs

Output

Enter number of years: 5 Show costs

Year	January	February	March	April	May	June	July	August	September	November	December	Total:
2014	352	404	669	574	342	101	525	334	798	213	831	5143
2013	790	146	233	728	248	137	133	778	344	150	357	4044
2012	121	333	985	877	272	901	117	236	369	352	250	4813
2011	307	977	321	761	583	282	65	548	86	752	269	4951
2010	891	210	575	188	690	927	313	506	272	780	57	5409

Problem 4. **Find Primes in Range**

Write a PHP script **PrimesInRange.php** that receives **two numbers** – **start** and **end** – from an **input field** and displays all numbers in that range as a **comma-separated list**. **Prime** numbers should be **bolded**. Styling the page is *optional.* Examples:

Input / Output

Starting Index: 2 End: 29 Submit

2, **3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20

Input / Output

Starting Index: 233 End: 651 Submit

233, 234, 235, 236, 237, 238, **239**, 240, **241**, 242, 243, 244, 245, 246, 247, 248, 249, 250, **251**, 252, 253, 254, 255, 256, **257**, 258, 259, 260, 261, 262, **263**, 264, 265, 266, 267, 268, **269**, 270, **271**, 272, 273, 274, 275, 276, **277**, 278, 279, 280, **281**, 282, **283**, 284, 285, 286, 287, 288, 289, 290, 291, 292, **293**, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, **307**, 308, 309, 310, **311**, 312, **313**, 314, 315, 316, **317**, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, **331**, 332, 333, 334, 335, 336, **337**, 338, 339, 340, 341, 342, 343, 344, 345, 346, **347**, 348, **349**, 350, 351, 352, **353**, 354, 355, 356, 357, 358, **359**, 360, 361, 362, 363, 364, 365, 366, **367**, 368, 369, 370, 371, 372, **373**, 374, 375, 376, 377, 378, **379**, 380, 381, 382, **383**, 384, 385, 386, 387, 388, **389**, 390, 391, 392, 393, 394, 395, 396, **397**, 398, 399, 400, **401**, 402, 403, 404, 405, 406, 407, 408, **409**, 410, 411, 412, 413, 414, 415, 416, 417, 418, **419**, 420, **421**, 422, 423, 424, 425, 426, 427, 428, 429, 430, **431**, 432, **433**, 434, 435, 436, 437, 438, **439**, 440, 441, 442, **443**, 444, 445, 446, 447, 448, **449**, 450, 451, 452, 453, 454, 455, 456, **457**, 458, 459, 460, **461**, 462, **463**, 464, 465, 466, **467**, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, **479**, 480, 481, 482, 483, 484, 485, 486, **487**, 488, 489, 490, **491**, 492, 493, 494, 495, 496, 497, 498, **499**, 500, 501, 502, **503**, 504, 505, 506, 507, 508, **509**, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, **521**, 522, **523**, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, **541**, 542, 543, 544, 545, 546, **547**, 548, 549, 550, 551, 552, 553, 554, 555, 556, **557**, 558, 559, 560, 561, 562, **563**, 564, 565, 566, 567, 568, **569**, 570, **571**, 572, 573, 574, 575, 576, **577**, 578, 579, 580, 581, 582, 583, 584, 585, 586, **587**, 588, 589, 590, 591, 592, **593**, 594, 595, 596, 597, 598, **599**, 600, **601**, 602, 603, 604, 605, 606, **607**, 608, 609, 610, 611, 612, **613**, 614, 615, 616, **617**, 618, **619**, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, **631**, 632, 633, 634, 635, 636, 637, 638, 639, 640, **641**, 642, **643**, 644, 645, 646, **647**, 648, 649, 650, 651

Problem 5. **Sum of Digits**

Write a PHP script **SumOfDigits.php** which receives a **comma-separated list of integers** from an **input form** and creates a **two-column table**. The first column should contain **each of the values** from the input. The second column should contain the **sum of the digits of each value**. If the value is not an integer number, print "**I cannot sum that**". Styling the page is *optional.* Example:

Input / Output	Input string: 1111, 2222, 3333, 4444, asdf Submit 1111 4 2222 8 3333 12 4444 16 asdf I cannot sum that

Problem 6. **Modify String**

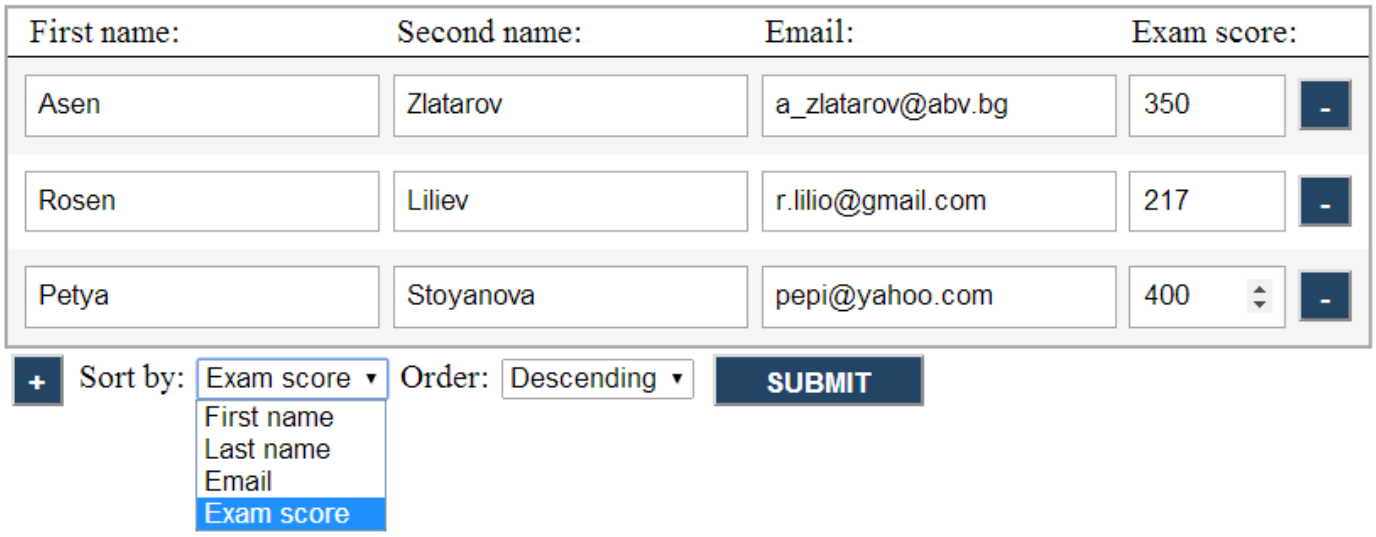
Write a PHP script **StringModifier.php** which receives a **string** from an **input form** and **modifies it** according to the selected option (radio button). You should support the following operations: **palindrome check**, **reverse string**, **split** to extract leters only, **hash** the string with the default PHP hashing algorithm, **shuffle** the string characters randomly. The result should be displayed right under the input field. Styling the page is *optional.* Think about which of the modification can be achieved with already built-in functions in PHP. Where necessary, write your own algorithms to modify the given string. Hint**:** Use the **crypt()** function for the "Hash String" modification. Examples:

Input / Output	Hello! ◉ Check Palindrome ○ Reverse String ○ Split ○ Hash String ○ Shuffle String Submit Hello! is not a palindrome!
Input / Output	PHP is awesome! ○ Check Palindrome ◉ Reverse String ○ Split ○ Hash String ○ Shuffle String Submit !emosewa si PHP
Input / Output	I can split that ○ Check Palindrome ○ Reverse String ◉ Split ○ Hash String ○ Shuffle String Submit I c a n s p l i t t h a t
Input / Output	programming ○ Check Palindrome ○ Reverse String ○ Split ◉ Hash String ○ Shuffle String Submit $2y$10$uo3cM4qEs.Ip4AT.WnyV6u5BCVWPIcbl1eWIfwzFiX/LZOg5ueVzC
Input / Output	Relax! Take it easy! ○ Check Palindrome ○ Reverse String ○ Split ○ Hash String ◉ Shuffle String Submit ! elteay Ra xeas!iTk

Problem 7. ** **Student Sorting**

Write a PHP program **StudentSorting.php** that receives data about several students from an **input form** (**first name**, **last name**, **email** and **grade**) and prints it as an HTML table. The user should be able to dynamically **add/remove** entries via the **+/-** buttons. The data can be sorted by 4 criteria: **First name**, **Last name**, **Email** and **Exam score**. The sorting can be done in **ascending/descending** order. The result should be printed as a table. The average exam score should be printed on the last row. (See the example below.) Styling the page is *optional*. Semantic HTML is required. (Hint: Use objects to store the data.)

Form



Result

First name	Last name	Email	Exam score
Petya	Stoyanova	pepi@yahoo.com	400
Asen	Zlatarov	a_zlatarov@abv.bg	350
Rosen	Liliev	r.lilio@gmail.com	217
Average Score:			**322**