2 Глава

1. Согласно данным Росстата от 24.07.2018 "Численность населения Российской Федерации по муниципальным образованиям на 1 января 2018 года"

|  |  |
| --- | --- |
| Новосибирск | 1612833 |
| Астрахань | 533925 |
| Санкт-Петербург | 5351935 |
| Екатеринбург | 1468833 |
| Москва | 12 506 468 |
| Нижний Новгород | 1259013 |

Определите один массив (или ряд массивов), хранящий местоположение и население перечисленных выше городов. Выведите на экран таблицу со сведениями о местоположении и населении, а также общее население всех городов.

Дополните выполнение задания в предыдущем упражнении таким образом, чтобы строки в результирующей таблице были упорядочены сначала по населению (в порядке убывания), а затем по названиям городов (в алфавитном порядке).

* 1. Согласно данным Росстата от 24.07.2018 "Численность населения Российской Федерации по муниципальным образованиям на 1 января 2018 года"

|  |  |  |
| --- | --- | --- |
| Астраханская область |  |  |
|  | Астрахань | 533925 |
|  | Нариманов | 50000 |
|  | Ахтубинск | 75000 |
| Московская область |  |  |
|  | Москва | 12506468 |
|  | Раменское | 150000 |
|  | Люберцы | 200000 |
| Ленинградская область |  |  |
|  | Санкт-Петербург | 5351935 |
|  | Волхов | 235000 |
|  | Выборг | 567000 |

Определите один список, хранящий местоположение и население перечисленных выше областей и городов. Выведите на экран таблицу со сведениями о местоположении и населении, а также общую численность по областям и численность всех городов.

4 Глава

|  |
| --- |
| 2. Создайте переменные $a, $b и присвойте им положительные числа. Присвойте частное от деления $a/$b третьей переменной $d. Используя условный оператор if, осуществите схему вывода на экран значения переменной $d только в том случае, если оно является положительным числом. Проверьте работу скрипта, после чего измените одно из значений переменных на отрицательное число и снова запустите скрипт. |
| 3. Создайте переменные $a, $b и присвойте им положительные числа. Присвойте частное от деления $a/$b третьей переменной $d. Используя условный оператор if, осуществите схему вывода на экран значения переменной $d только в том случае, если оно является положительным числом. Если число окажется отрицательным или равным нулю, должно выводиться соответствующее предупреждение, а не значение переменной. Проверьте работу скрипта, после чего измените одно из значений переменных $a или $b на отрицательное число и снова запустите скрипт. |
| 4. Создайте переменные $a, $b и присвойте им положительные числа. Присвойте частное от деления $a/$b третьей переменной $d. Используя условный оператор if в связке с else if, осуществите схему вывода на экран значения переменной $d только в том случае, если оно является положительным числом. Если число окажется отрицательным или равным нулю, то для каждого случая должно выводиться соответствующее предупреждение. Проверьте работу скрипта, после чего измените одно из значений переменных $a или $b на отрицательное число и запустите скрипт снова. Чтобы проверить работу скрипта для нулевого значения, измените значение переменной $a на ноль. |
| 5. Реализуйте условие задачи 4 при помощи оператора switch. |
| 6. Создайте переменную и присвойте ей массив, состоящий из пяти элементов, например, чисел или строк. Создайте еще одну переменную, которой присвойте какое-нибудь значение. Теперь создайте конструкцию switch, которая будет определять, есть такой элемент в массиве или нет и выводить соответствующее сообщение. Для упрощения кода используйте операторы case, идущие друг за другом и содержащие метки в виде значений элементов массива, но не прерывающиеся операторами break. Не забудьте про условие по умолчанию, которое будет выводить сообщение, если элемента в массиве нет. Протестируйте конструкцию, присваивая второй переменной различные значения. |
| 7. Создайте переменную $a и присвойте ей ноль в качестве стартового значения. Выведите на экран при помощи цикла while цифры от 1 до 5, использовав для этого операцию префиксного инкремента переменной $a. Сбросьте значение переменной и сделайте тоже самое, но при помощи цикла do/while. |
| 8. Создайте переменную $a и присвойте ей массив, состоящий из пяти элементов. Выведите все элементы массива на экран сперва при помощи цикла while, а затем и do/while. |
| 9. Выведите все элементы массива из предыдущей задачи на экран при помощи цикла for. Обратите внимание на компактность кода. |
| 10. Используйте цикл for для вывода на экран целых положительных чисел не превышающих 50. Прервите цикл на 10 итерации при помощи внутреннего условного оператора if и оператора break. |
| 11. Выведите на экран таблицу умножения. Используйте внешний и внутренний циклы for. Чтобы таблица имела привычный для нас 'квадратный' вид, в конце каждой итерации внешнего цикла задайте вывод разрыва строки. |
| 12. Создайте массив, состоящий из пяти элементов. Выведите на экран значения элементов массива, использовав для этого цикл foreach в двух вариантах: с выводом только значений и с выводом как ключей, так и значений соответствующих элементов. |
| 13. Создайте массив, состоящий из пяти элементов в виде целых чисел. Увеличьте значения всех элементов массива в два раза, использовав для этого цикл foreach. Выведите на экран новые значения элементов массива. |
| 14. Выведите на экран при помощи цикла for положительные целые числа от 1 до 10. Затем используйте оператор continue, чтобы исключить из ряда числа 3,4,5. Попробуйте разместить оператор continue до и после инструкции вывода значений на экран и объясните результат. |
| 15. Используйте условие задачи 11, но исключите из таблицы столбик умножения на 5. Для этого прервите внутренний цикл оператором continue при $k==5. |
| 16. Задайте бесконечный цикл for, который будет выводить квадраты целых положительных чисел. Цикл будет бесконечным, если второй параметр в условии будет отсутствовать. Если квадрат числа превысит 70, прервите цикл при помощи оператора goto, который будет осуществлять переход к метке после цикла для вывода соответствующего сообщения. |
| 17. Разработайте программу, которая из чисел 20 .. 45 находила те, которые делятся на 5 и найдите сумму этих чисел. |
| Экстремальное программирование этап 1  Выведите на экран величины температур в пределах от -50 до 50 градусов по Фаренгейту и эквивалентные им величины температур в градусах Цельсия.  По температурной шкале Фаренгейта вода замерзает при температуре 32 градуса и закипает при 212 градусах. А по температурной шкале Цельсия вода замерзает при температуре О градусов и закипает при 100 градусах.  *Таким образом, для преобразования температуры по Фаренгейту в температуру по Цельсию следует вычесть из ее величины 32, умножить полученную разность на 5 и разделить на 9. А для преобразования температуры по Цельсию в температуру по Фаренгейту следует умножить ее величину на 9, разделить полученный результат на 5 и прибавить 32.*  Проверка гипотезы Сиракуз  Гипотеза Сиракуз гласит, что любое натуральное число сводится к единице при следующих действиях над ним: а) если число четное, то разделить его пополам, б) если число нечетное, то умножить его на 3, прибавить 1 и результат разделить на 2. Над вновь полученным числом вновь повторить действия a) или б) в зависимости от четности числа. Рано или поздно число станет равным 1.  Напишем программу, проверяющую гипотезу Сиракуз. Пусть пользователь вводит любое натуральное число. Будем выполнять в цикле над ним вышеуказанные действия, пока оно не станет равным 1. Если гипотеза верна, то рано или поздно цикл закончится, а если нет - то произойдет зацикливание программы.  Пример:  100  50 25 38 19 29 44 22 11 17 26 13 20 10 5 8 4 2 1 |
|  |

**5. Функции**

1. Создайте пользовательскую функцию, которая принимает два аргумента, а возвращает их произведение. Вызовите функцию, передав ей в качестве аргументов два числа и выведите на экран результат.

2. Создайте три переменные и присвойте им числовые значения. Выведите на экран их произведение. Теперь создайте пользовательскую функцию, принимающую два аргумента по ссылке и один аргумент по значению. Функция должна присваивать переменным другие числовые значения. Вызовите функцию, после чего снова выведите на экран произведение всех переменных. Объясните результат.

3. Создайте две переменные и присвойте им числовые значения. Теперь создайте пользовательскую функцию, принимающую два аргумента со значениями по умолчанию и выводящую произведение своих аргументов. Вызовите функцию, передав ей сперва в качестве аргументов значения переменных, затем значение только одной из переменных и, наконец, вообще без аргументов.

4 Создайте пользовательскую функцию, принимающую аргументы в массив переменной длины и выводящую их затем на экран. Используйте для доступа к элементам массива цикл foreach. Вызовите функцию, передав ей в качестве значения две строки и число.

5 Создайте две переменные $sum и $product и присвойте им стартовые значения, равные 0 и 1. Создайте пользовательскую функцию, принимающую два аргумента по ссылке, а все остальные по значению в массив переменной длины. Функция должна вычислять сумму всех аргументов, переданных в массив переменной длины, и присваивать ее первому аргументу. Второму аргументу она должна присваивать произведение всех аргументов, переданных в массив переменной длины. Возвращать функция должна массив, состоящий из первых двух аргументов, переданных функции (т.е. сумму и произведение). Используйте для доступа к элементам массива цикл foreach. Вызовите функцию, передав ей в качестве первых двух аргументов наши переменные, а также три числа. Выведите на экран значения возвращенного массива, а также значения наших переменных. Объясните результат.

6 Создайте пользовательскую функцию, которая будет выводить на экран сумму первых двух переданных ей аргументов. Количество передаваемых функции аргументов не указывайте. Если функция будет получать более двух аргументов, она должна выдавать соответствующее предупреждение и не выводить сумму. Вызовите функцию, передав ей в качестве аргументов массив из двух чисел и развернув его в простые аргументы при помощи '...'. Повторите тоже самое для массива из трех чисел. Напомним, что количество переданных функции элементов можно узнать при помощи функции func\_num\_args(), а получить доступ к ним можно при помощи функции func\_get\_arg(n).

7 Задайте режим строгой типизации (используйте инструкцию declare(strict\_types=1);), после чего создайте пользовательскую функцию, которая будет принимать два целочисленных аргумента и выводить на экран их сумму. Вызовите функцию, передав ей в качестве аргументов сперва два целых числа, а потом одно из них в виде строки. Закомментируйте строку с инструкцией строгой типизации и перезапустите скрипт. Объясните результат. Показать решение.

8 Задайте режим строгой типизации (используйте инструкцию declare(strict\_types=1);), после чего создайте пользовательскую функцию my\_func(), которая будет принимать два целочисленных аргумента и возвращать их произведение. Создайте переменную $count\_apples и присвойте ей строку с именем функции. Обратитесь к функции через переменную и выведите на экран общую массу яблок зная, что имеется 23 корзины по 7 кг яблок в каждой. Показать решение.

**Калькулятор**. Напишите функцию, которая принимает три аргумента: первые два – числовые, третий – символ, обозначающий операцию. **Функция должна возвращать** результат выполнения третьего оператора над первым и вторым.

26. Создайте класс, в котором задайте константу, общедоступные свойство и метод класса. Метод класса должен выводить какое-нибудь сообщение. Создайте экземпляр класса. Вызовите метод класса, а также выведите на экран сумму значений константы класса и его свойства. Не забудьте, что для доступа к константе класса нужно использовать двойное двоеточие '::', а не объектный оператор '->'.

27. Имеется 7 корзин. Нужно разработать класс, который будет подсчитывать общую массу яблок в зависимости от вместимости корзин. Для этого создайте сам класс. Затем создайте константу класса, которой присвойте количество корзин. Объявите свойство класса, но не инициализируйте его. Свойство будет отвечать за вместимость корзины в килограммах. И, наконец, объявите метод, который будет принимать в качестве аргумента вместимость корзины, т.е. значение соответствующего свойства, а возвращать произведение количества корзин (значение нашей константы) на вместимость одной корзины. После оформления класса, создайте его экземпляр и выведите на экран количество корзин. Не забудьте, что для доступа к константе класса нужно использовать двойное двоеточие '::', а не объектный оператор '->'. Затем присвойте свойству класса положительное число и выведите на экран общую массу яблок, которую могут вместить все корзины, использовав для этого оператор echo и вызов метода класса.

Формы.

1. Спросите **имя** пользователя с помощью формы. Результат запишите в переменную **$name**. Выведите на экран фразу **'Привет, %Имя%'**
2. Спросите у пользователя **имя**, **возраст**, а также попросите его ввести **сообщение** (его сделайте в **textarea**). Выведите эти данные на экран в **формате**, приведенном под данной задачей. Позаботьтесь о том, чтобы пользователь **не мог вводить теги** (просто удаляйте их, функция [strip\_tags](http://code.mu/php/string/strip_tags.html), удалит теги) и таким образом сломать сайт.

Привет, Дмитрий, 25 лет.  
Твое сообщение: ...

1. Спросите **возраст** пользователя. Если форма была **отправлена** и **введен** возраст, то выведите его на экран, а форму уберите. Если же форма не была отправлена (это будет при **первом** заходе на страницу) - просто покажите ее.
2. Спросите у пользователя **логин** и **пароль** (в браузере должен быть звездочками). Сравните их с логином **$login** и паролем **$pass**, хранящихся в файле. Если все верно - выведите 'Доступ разрешен!', в противном случае - 'Доступ запрещен!'. Сделайте так, чтобы скрипт обрезал концевые пробелы в строках, которые ввел пользователь.