//Обоснования в комментариях в коде

1. Программа выведет 8.52.333333333333332111

$x = 2.5;

$y = 3;

print ($x + 2\*$y); # на экран выводится 8.5

print (7 / $y); # на экран выводится 2.33333333333333, по умолчанию длина 16 знаков у типа с плавающей точкой

print int (7 / $y); # на экран выводится 2 т.к. отбрасывается дробная часть

print (7 % $y); # на экран выводится 1, это остаток от деления

print (7.5 % $y); # на экран выводится 1, выводится целочисленный остаток от деления

print (7.5 % $x); # на экран выводится 1, выводится целочисленный остаток от деления

2. Зависит от уровня вывода ошибок. Либо ошибку использования неинициализированной перменной, либо следующий текст:

Invest only and become millionaire!Invest only $1 and become millionaire!

В случае с двойными кавычками компилятор принимает $1 за переменную и пытается подставить её значение, но она неинициализирована и со строкой конкатенируется пустая строка.

3. Вывести данные в формате: Total time: 3h 14m

$hours=3;

$minutes=14;

print("Total time ${hours}h ${minutes}m")

4. Применить к указанным данным операции сложения, конкатенации и повторения, меняя местами

операнды в последних двух операциях. Пояснить, как и почему именно так происходит

предложенный ниже вывод результатов функцией print

$s = "3";

$n = 2;

$res1 = $s + $n;

$res2 = $s . $n;

$res3 = $n . $s;

$res4 = $s x $n;

$res5 = $n x $s;

print "\n$res1, $res2, $res3, $res4, $res5\t"; #5, 32, 23, 33, 222

# при сложении строки преобразуются в числа, при конкатенации числа в строки, при повторении то что справа приводится к числу, слева к строке

print $res1, $res2, $res3, $res4, $res5; #5322333222 просто вывели всё подряд

$, = " "; #$, - Разделитель полей выходных записей для функции print (между параметрами)

$\ = "\n"; #$\ - Разделитель выходных записей для функции print (добавляется в конец вывода при вызове print)

print $res1, $res2, $res3, $res4, $res5; #5 32 23 33 222

# т.к. добавили разделитель между параметрами и в конце вывода print вставляются пробелы и символы новой строки соответственно

print $res1, $res2, $res3, $res4, $res5; #5 32 23 33 222

print "==================================="; #=================================== и в конце символ новой строки

$n = "2";

$m = "0.1";

$res1 = $n + $m;

$res2 = $n . $m;

$res3 = $m . $n;

$res4 = $n x $m;

$res5 = $m x $n;

print $res1,$res2, $res3,$res4,$res5; #2.1 20.1 0.12 0.10.1

# 2.1 привелись к числам и сложились, далее два результата конкатенации строк, далее ничего потому что 0.1 привели к целому числу 0 и умножили на него количество

# выводов строки "2", дале 0.10.1 потому что строку "0.1" продублировали 2 раза (строка "2" привелась к числу как правый операнд дублирования строк)

=pod Вывод:

5, 32, 23, 33, 222 53223332225 32 23 33 222

5 32 23 33 222

===================================

2.1 20.1 0.12 0.10.1

=cut

5. Продемонстрировать использование всех операций строкового сравнения для указанных ниже

данных, используя унаследованный от языка С условный оператор if ... else. Результат

обосновать.

$a = "acb"; # ASCII: a - 97, c - 99, b - 98

$b = "ab3"; # ASCII: a - 97, b - 98, 3 - 51

print "Value of \$a = $a and value of \$b = $b\n";

if($a eq $b) {

print "\$a eq \$b is true\n";

} else {

print "\$a eq \$b is false\n"; # false, a == a, c != b

}

if($a gt $b) {

print "\$a gt \$b is true\n"; # true, a == a, c > b

} else {

print "\$a gt \$b is false\n";

}

if($a lt $b) {

print "\$a lt \$b is true\n";

} else {

print "\$a lt \$b is false\n"; # false, a == a, c > b

}

if($a ge $b) {

print "\$a ge \$b is true\n"; # true, a == a, c > b

} else {

print "\$a ge \$b is false\n";

}

if($a le $b) {

print "\$a le \$b is true\n";

} else {

print "\$a le \$b is false\n"; # false, a == a, c > b

}

if($a ne $b) {

print "\$a ne \$b is true\n"; #true, a == a, c != b

} else {

print "\$a ne \$b is false\n";

}

print "\$a cmp \$b = " . ($a cmp $b); # 1, a == a, c > b значит 1, т.к. больше