

1) Заменить в индексном (простом) массиве все элементы, в которых встречается цифра 2.

replace.php

```
<?php

const n = 99;
$a = -1;
$flag = False;
$arr = array();

for ($i = 0; $i <= n; $i++) {
    $arr[$i] = rand();
}

foreach ($arr as &$value) {
    if ($flag == True) {
        $value = $a;
        $flag = False;
    } else {
        $b = (string) $value;
        $pos = strpos($b, '2');
        if ($pos !== false) {
            $flag = True;
        }
    }
}

unset($value);

?>
```

1.2) Вставить \$a в индексный (простой) массив целых чисел после всех элементов, в которых есть цифра 2. Новый массив создавать нельзя. Использовать функцию array_splice нельзя.

insert.php

Ошибка: функция key(\$arr) не возвращает индекс текущего элемента, поэтому все вставки производятся в начало массива.

```
<?php

$a = -1;
$flag = False;
$arr = array();
$arr_dim = rand(1, 100);

for ($i = 0; $i <= $arr_dim-1; $i++) {
    $arr[$i] = rand();
}
```

```

foreach ($arr as &$value) {
    if ($flag == True) {
        $j = $arr_dim - 1;
        while ($j >= $index) {
            $arr[$j+1] = $arr[$j];
            $j--;
        }
        $arr[$index] = $a;
        $arr_dim++;
        $flag = False;
    } else {
        $b = (string) $value;
        $pos = strpos($b, '2');
        if ($pos !== false) {
            $index = key($arr);
            $flag = True;
        }
    }
}

unset($value);

?>

```

2) Создать PHP-страницу upload.php с формой загрузки CSV-файла. В CSV-файле должны быть 2 столбца: название файла, содержимое. Рядом с файлом upload.php требуется создать папку /upload/ и создать в ней файлы, прочитав CSV-файл.

upload.php

Ошибка: тип text/csv через функцию input не реализуется.

```

<!DOCTYPE html>
<html>
<head>
<title>CSV file upload</title>
</head>
<body>
<?php
if ($_FILES && $_FILES["filename"]["error"] == UPLOAD_ERR_OK)
{
    $name = $_FILES["filename"]["name"];
    move_uploaded_file($_FILES["filename"]["tmp_name"], $name);
    $dir = "/upload/";
    mkdir($dir, 0755);
    $row = 1;
    if (($handle = fopen($_FILES["filename"]["name"], "r")) !== FALSE) {
        while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
            $fnw = fopen($dir . $data[0], "w");
            fwrite($fnw, $data[1]);
            fclose($fnw);
        }
    }
}

```

```

        $row++;
    }
}
fclose($handle);
echo "Uploaded file parsed, new files produced.";
}
?>
<h2>Uploading a File</h2>
<form method="post" enctype="multipart/form-data">
Select CSV file: <input type="text/csv" name="filename" /><br /><br />
<input type="submit" value="Upload" />
</form>
</body>
</html>

```

3) Вашей команде поставлена задача – создать онлайн версию своей системы управления проектами (по аналогии с Redmine, JIRA).

Вам нужно разработать структуру таблиц с необходимыми полями, связи между ними, продемонстрировать и рассказать о причинах выбора того или иного типа поля или связи между полями.

Вы можете добавлять к условию задачи любые дополнительные вводные (сущности, допущения, ограничения), которые посчитаете необходимыми, но в структуре БД обязательно должны быть отражены следующие главные информационные сущности: "Проект", "Задача", "Исполнитель", а у каждой задачи обязательно должна быть возможность указать планируемый срок исполнения, фактический срок исполнения, планируемую трудоемкость, фактическую трудоемкость.

см. CMS Project.pdf, CMS Project ERD.pdf

3.1) Нарисовать ER-диаграмму БД для записи результатов соревнований бегунов марафона. В БД должны храниться Спортсмены, Соревнования, Результаты (спортсмена по соревнованию) и Список призеров (до 3-х на соревновании)

см. ERD.pdf

4) Напишите SQL-запрос для создания таблицы спортсменов (подберите подходящие типы данных):

- ФИО
- e-mail
- телефон (только российские номера мобильных)
- дата рождения
- возраст,
- дата и время создания записи
- номер паспорта,
- среднее место на соревнованиях
- биография
- видеопрезентация.

USE Marathons_Results

```
CREATE TABLE Sportsmen
(Id INT PRIMARY KEY IDENTITY,
 Surname VARCHAR(80) NOT NULL,
 FirstName VARCHAR(60) NOT NULL,
 LastName VARCHAR(70) NOT NULL,
 PS_Series SMALLINT NOT NULL,
 PS_Number INT NOT NULL,
 PS_Granted VARCHAR(150) NOT NULL,
 Birthdate DATE NOT NULL,
 Age TINYINT CHECK(Age>0 AND Age<60) NOT NULL,
 Phone VARCHAR(15) UNIQUE,
 Email VARCHAR(50) UNIQUE,
 Biography CHAR(8000) NOT NULL,
 Video VARBINARY(MAX),
 CompetsNumber SMALLINT,
 AveragePlace TINYINT,
 Rate VARCHAR(30),
 Record_DT TIMESTAMP
)
```

4.1) Простенький запрос с выборкой строк из таблицы, упорядоченных по определённому принципу (топ 5 ФИО спортсменов, больше остальных посетивших соревнований).

```
SELECT * FROM Sportsmen ORDER BY CompetsNumber DESC FETCH FIRST 5 ROWS ONLY;
```

Альтернативный вариант:

```
SELECT Surname, FirstName, LastName, CompetsNumber FROM Sportsmen ORDER BY CompetsNumber DESC FETCH FIRST 5 ROWS ONLY;
```