```php
#!/usr/bin/env php
<?php

    // ask user for an integer
    $n = readline("I'd like an integer please: ");

    // analyze user's input
    if ($n > 0)
    {
        printf("You picked a positive number!\n");
    }
    else if ($n == 0)
    {
        printf("You picked zero!\n");
    }
    else
    {
        printf("You picked a negative number!\n");
    }

?>
```

```
<?php

    /**
     * froshims-0.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-0.php.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body style="text-align: center;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-0.php" method="post">
            Name: <input name="name" type="text"/>
            <br/>
            <input name="captain" type="checkbox"/> Captain?
            <br/>
            <input name="comfort" type="radio" value="less"/> Less Comfortable
            <input name="comfort" type="radio" value="more"/> More Comfortable
            <br/>
            Dorm:
            <select name="dorm">
                <option value=""></option>
                <option value="Apley Court">Apley Court</option>
                <option value="Canaday">Canaday</option>
                <option value="Grays">Grays</option>
                <option value="Greenough">Greenough</option>
                <option value="Hollis">Hollis</option>
                <option value="Holworthy">Holworthy</option>
                <option value="Hurlbut">Hurlbut</option>
                <option value="Lionel">Lionel</option>
                <option value="Matthews">Matthews</option>
                <option value="Mower">Mower</option>
                <option value="Pennypacker">Pennypacker</option>
                <option value="Stoughton">Stoughton</option>
                <option value="Straus">Straus</option>
                <option value="Thayer">Thayer</option>
                <option value="Weld">Weld</option>
                <option value="Wigglesworth">Wigglesworth</option>
            </select>
            <br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

```
<?php

    /**
     * froshims-1.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-1.php.
     *
     * Demonstrates Bootstrap (http://getbootstrap.com/).
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"/>
        <title>Frosh IMs</title>
    </head>
    <body style="margin: 20px;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-1.php" method="post">
            <fieldset>
                <label>Name</label>
                <input name="name" type="text"/>
                <label class="checkbox">
                    <input name="captain" type="checkbox"/> Captain?
                </label>
                <label class="radio">
                    <input name="comfort" type="radio" value="less"/> Less Comfortable
                </label>
                <label class="radio">
                    <input name="comfort" type="radio" value="more"/> More Comfortable
                </label>
                <label>
                    <select name="dorm">
                        <option value=""></option>
                        <option value="Apley Court">Apley Court</option>
                        <option value="Canaday">Canaday</option>
                        <option value="Grays">Grays</option>
                        <option value="Greenough">Greenough</option>
                        <option value="Hollis">Hollis</option>
                        <option value="Holworthy">Holworthy</option>
                        <option value="Hurlbut">Hurlbut</option>
                        <option value="Lionel">Lionel</option>
                        <option value="Matthews">Matthews</option>
                        <option value="Mower">Mower</option>
                        <option value="Pennypacker">Pennypacker</option>
                        <option value="Stoughton">Stoughton</option>
                        <option value="Straus">Straus</option>
                        <option value="Thayer">Thayer</option>
                        <option value="Weld">Weld</option>
                        <option value="Wigglesworth">Wigglesworth</option>
                    </select>
                </label>
                <button class="btn btn-default" type="submit">Register</button>
            </fieldset>
        </form>
    </body>
</html>
```

```
<?php

    /**
     * froshims-2.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-2.php.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body style="text-align: center;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-2.php" method="post">
            Name: <input name="name" type="text"/>
            <br/>
            <input name="captain" type="checkbox"/> Captain?
            <br/>
            <input name="comfort" type="radio" value="less"/> Less Comfortable
            <input name="comfort" type="radio" value="more"/> More Comfortable
            <br/>
            Dorm:
            <select name="dorm">
                <option value=""></option>
                <option value="Apley Court">Apley Court</option>
                <option value="Canaday">Canaday</option>
                <option value="Grays">Grays</option>
                <option value="Greenough">Greenough</option>
                <option value="Hollis">Hollis</option>
                <option value="Holworthy">Holworthy</option>
                <option value="Hurlbut">Hurlbut</option>
                <option value="Lionel">Lionel</option>
                <option value="Matthews">Matthews</option>
                <option value="Mower">Mower</option>
                <option value="Pennypacker">Pennypacker</option>
                <option value="Stoughton">Stoughton</option>
                <option value="Straus">Straus</option>
                <option value="Thayer">Thayer</option>
                <option value="Weld">Weld</option>
                <option value="Wigglesworth">Wigglesworth</option>
            </select>
            <br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

```
<?php

    /**
     * froshims-3.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-3.php.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body style="text-align: center;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-3.php" method="post">
            Name: <input name="name" type="text"/>
            <br/>
            <input name="captain" type="checkbox"/> Captain?
            <br/>
            <input name="comfort" type="radio" value="less"/> Less Comfortable
            <input name="comfort" type="radio" value="more"/> More Comfortable
            <br/>
            Dorm:
            <select name="dorm">
                <option value=""></option>
                <option value="Apley Court">Apley Court</option>
                <option value="Canaday">Canaday</option>
                <option value="Grays">Grays</option>
                <option value="Greenough">Greenough</option>
                <option value="Hollis">Hollis</option>
                <option value="Holworthy">Holworthy</option>
                <option value="Hurlbut">Hurlbut</option>
                <option value="Lionel">Lionel</option>
                <option value="Matthews">Matthews</option>
                <option value="Mower">Mower</option>
                <option value="Pennypacker">Pennypacker</option>
                <option value="Stoughton">Stoughton</option>
                <option value="Straus">Straus</option>
                <option value="Thayer">Thayer</option>
                <option value="Weld">Weld</option>
                <option value="Wigglesworth">Wigglesworth</option>
            </select>
            <br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

```
<?php

    /**
     * register-0.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Dumps contents of $_POST.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        <pre><?php print_r($_POST); ?></pre>
    </body>
</html>
```

```php
<?php

    /**
     * register-1.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.  Redirects
     * user to froshims-1.php upon error.
     */

    // validate submission
    if (empty($_POST["name"]) || empty($_POST["comfort"]) || empty($_POST["dorm"]))
    {
        header("Location: froshims-1.php");
        exit;
    }

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        You are registered!  (Well, not really.)
    </body>
</html>
```

```
<?php

    /**
     * register-2.php
     *
     * Computer Science 50
     * David J. Malan
     *
     * Implements a registration form for Frosh IMs.  Informs user of
     * any errors.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        <?php if (empty($_POST["name"]) || empty($_POST["comfort"]) || empty($_POST["dorm"])): ?>
            You must provide your name, comfort, and dorm!  Go <a href="froshims-2.php">back</a>.
        <?php else: ?>
            You are registered!  (Well, not really.)
        <?php endif ?>
    </body>
</html>
```

```php
<?php

    /**
     * register-3.php
     *
     * Computer Science 50
     * David J. Malan
     *
     * Implements a registration form for Frosh IMs.  Reports registration
     * via email.  Redirects user to froshims-3.php upon error.
     */

    // require PHPMailer
    require("libphp-phpmailer/class.phpmailer.php");

    // validate submission
    if (!empty($_POST["name"]) && !empty($_POST["comfort"]) && !empty($_POST["dorm"]))
    {
        // instantiate mailer
        $mail = new PHPMailer();

        // use SMTP
        $mail->IsSMTP();
        $mail->Host = "smtp.gmail.com";
        $mail->Password = "crimson50";
        $mail->Port = 587;
        $mail->SMTPAuth = true;
        $mail->SMTPSecure = "tls";
        $mail->Username = "jharvard@cs50.net";

        // set From:
        $mail->SetFrom("jharvard@cs50.net");

        // set To:
        $mail->AddAddress("jharvard@cs50.net");

        // set Subject:
        $mail->Subject = "registration";

        // set body
        $mail->Body =
            "This person just registered:\n\n" .
            "Name: " . $_POST["name"] . "\n" .
            "Captain: " . $_POST["captain"] . "\n" .
            "Comfort: " . $_POST["comfort"] . "\n" .
            "Dorm: " . $_POST["dorm"];

        // send mail
        if ($mail->Send() == false)
        {
            print($mail->ErrInfo);
            exit;
        }
    }
    else
    {
        header("Location: froshims-3.php");
        exit;
    }
?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        You are registered!  (Really.)
    </body>
</html>
```

```
#!/usr/bin/env php
<?php

    printf("hello, world\n");

?>
```

```php
<?php

    /**
     * dictionary.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a dictionary in a (non-object-oriented) way that
     * mimics Problem Set 6's implementation in C.  However, an
     * object-oriented design would be better in PHP.
     */

    // size of dictionary
    $size = 0;

    // hash table
    $table = [];

    /**
     * Returns true if word is in dictionary else false.
     */
    function check($word)
    {
        global $table;
        if (isset($table[strtolower($word)]))
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    /**
     * Loads dictionary into memory.  Returns true if successful else false.
     */
    function load($dictionary)
    {
        global $table, $size;
        if (!file_exists($dictionary) && is_readable($dictionary))
        {
            return false;
        }
        foreach (file($dictionary) as $word)
        {
            $table[chop($word)] = true;
            $size++;
        }
        return true;
    }

    /**
     * Returns number of words in dictionary if loaded else 0 if not yet loaded.
     */
    function size()
    {
        global $size;
        return $size;
    }

    /**
     * Unloads dictionary from memory.  Returns true if successful else false.
     */
    function unload()
    {
        return true;
    }

?>
```

```php
#!/usr/bin/env php
<?php

    /***********************************************************************
     * speller.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a spell-checker.
     **********************************************************************/

    require("dictionary.php");

    // maximum length for a word
    // (e.g., pneumonoultramicroscopicsilicovolcanoconiosis)
    define("LENGTH", 45);

    // default dictionary
    define("WORDS", "/home/cs50/pset5/dictionaries/large");

    // check for correct number of args
    if ($argc != 2 && $argc != 3)
    {
        print("Usage: speller [dictionary] text\n");
        return 1;
    }

    // benchmarks
    $time_load = 0.0; $time_check = 0.0; $time_size = 0.0; $time_unload = 0.0;

    // determine dictionary to use
    $dictionary = ($argc == 3) ? $argv[1] : WORDS;

    // load dictionary
    $before = microtime(true);
    $loaded = load($dictionary);
    $after = microtime(true);

    // abort if dictionary not loaded
    if (!$loaded)
    {
        print("Could not load $dictionary.\n");
        return 1;
    }

    // calculate time to load dictionary
    $time_load = $after - $before;

    // try to open file
    $file = ($argc == 3) ? $argv[2] : $argv[1];
    $fp = fopen($file, "r");
    if ($fp === false)
    {
        print("Could not open $file.\n");
        return 1;
    }

    // prepare to report misspellings
    printf("\nMISSPELLED WORDS\n\n");

    // prepare to spell-check
    $word  = "";
    $index = 0; $misspellings = 0; $words = 0;

    // spell-check each word in file
    for ($c = fgetc($fp); $c !== false; $c = fgetc($fp))
    {
        // allow alphabetical characters and apostrophes (for possessives)
        if (preg_match("/[a-zA-Z]/", $c) || ($c == "'" && $index > 0))
        {
            // append character to word
            $word .= $c;
            $index++;

            // ignore alphabetical strings too long to be words
            if ($index >= LENGTH)
            {
                // consume remainder of alphabetical string
                while (($c = fgetc($fp)) !== false && preg_match("/[a-zA-Z]/", $c));

                // prepare for new word
                $index = 0; $word = "";
            }
        }

        // ignore words with numbers (like MS Word)
        else if (ctype_digit($c))
        {
            // consume remainder of alphabetical string
            while (($c = fgetc($fp)) !== false && preg_match("/[a-zA-z0-9]/", $c));

            // prepare for new word
            $index = 0; $word = "";
        }

        // we must have found a whole word
        else if ($index > 0)
        {
            // update counter
            $words++;

            // check word's spelling
            $before = microtime(true);
            $misspelled = !check($word);
            $after = microtime(true);

            // update benchmark
            $time_check += $after - $before;

            // print word if misspelled
            if ($misspelled)
            {
                print("$word\n");
                $misspellings++;
            }

            // prepare for next word
            $index = 0; $word = "";
        }
    }

    // close file
    fclose($fp);

    // determine dictionary's size
    $before = microtime(true);
    $n = size();
    $after = microtime(true);

    // calculate time to determine dictionary's size
    $time_size = $after - $before;

    // unload dictionary
    $before = microtime(true);
    $unloaded = unload();
    $after = microtime(true);

    // abort if dictionary not unloaded
    if (!$unloaded)
    {
        print("Could not load $dictionary.\n");
        return 1;
    }
    // calculate time to determine dictionary's size
    $time_unload = $after - $before;

    // report benchmarks
    printf("\nWORDS MISSPELLED:     %d\n", $misspellings);
    printf("WORDS IN DICTIONARY:  %d\n", $n);
    printf("WORDS IN TEXT:        %d\n", $words);
    printf("TIME IN load:         %.2f\n", $time_load);
    printf("TIME IN check:        %.2f\n", $time_check);
    printf("TIME IN size:         %.2f\n", $time_size);
    printf("TIME IN unload:       %.2f\n", $time_unload);
    printf("TOTAL TIME:           %.2f\n\n", $time_load + $time_check + $time_size + $time_unload);

?>
```

```
#!/usr/bin/env php
<?php

    $x = 2;
    printf("x is now %d\n", $x);
    printf("Cubing...\n");
    $x = cube($x);
    printf("Cubed!\n");
    printf("x is now %d\n", $x);

    /**
     * Cubes argument.
     */
    function cube($n)
    {
        return $n * $n * $n;
    }

?>
```

```php
<?php

    require("libphp-phpmailer/class.phpmailer.php");

    // open file
    $handle = fopen("TODO", "r");
    if ($handle == false)
    {
        print("could not open file\n");
        exit(1);
    }

    // array to hold email addresses
    $addresses = [];

    // iterate over rows in file
    while ($row = fgetcsv($handle))
    {
        // name is in first column
        $name = $row[0];

        // number is in third column
        $number = $row[2];

        // carrier is in fourth column
        $carrier = $row[3];

        // determine address
        if ($carrier == "AT&T")
        {
            array_push($addresses, "{$number}@txt.att.net");
        }
        else if ($carrier == "Sprint")
        {
            array_push($addresses, "{$number}@messaging.sprintpcs.com");
        }
        else if ($carrier == "T-Mobile")
        {
            array_push($addresses, "{$number}@tmomail.net");
        }
        else if ($carrier == "Verizon")
        {
            array_push($addresses, "{$number}@vtext.com");
        }
        else if ($carrier == "Virgin Mobile")
        {
           array_push($addresses, "{$number}@vmobl.com");
        }
    }

    // close file
    fclose($handle);

    // instantiate mailer
    $mail = new PHPMailer();

    // configure mailer
    // http://phpmailer.worxware.com/index.php?pg=methods
    // http://phpmailer.worxware.com/index.php?pg=properties
    // https://www.google.com/settings/u/0/security/lesssecureapps
    $mail->IsSMTP();
    $mail->Host = "smtp.gmail.com";
    $mail->Password = "TODO";
    $mail->Port = 587;
    $mail->SMTPAuth = true;
    $mail->SMTPDebug = 1;
    $mail->SMTPSecure = "tls";
    $mail->Username = "TODO";

    // set From:
    $mail->SetFrom("bot@cs50.net");

    // set body
    $mail->Body = "Miss you! love, CS50 Bot";

    // iterate over email addresses
    for ($i = 0, $n = count($addresses); $i < $n; $i++)
    {
        // add email address to To: field
        $mail->addAddress($addresses[$i]);

        // send email
        if ($mail->Send())
        {
            print("Sent text #{$i}.\n");
        }
        else
        {
            print($mail->ErrorInfo);
        }

        // clear To: field
        $mail->ClearAddresses();
    }

?>
```