Week 7 on Building Applications in PHP

In this next assignment, we'll build code to reverse an MD5 hash using a brute force technique where we simply 'forward hash' all possible combinations of characters in strings. This would be similar to a situation where an e-commerce site stored hashed passwords in its database and we somehow have gotten our hands on the database contents and we want to take the hashed password and determine the actual plaintext passwords.

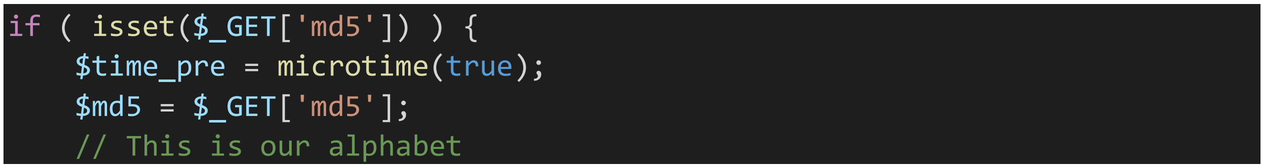
This following is a list of people, and their hashed PIN values.

email	**pin**	**hash_pin**
csev@umich.edu	????	0bd65e799153554726820ca639514029
nabgilby@umich.edu	????	aa36c88c27650af3b9868b723ae15dfc
pconway@umich.edu	????	1ca906c1ad59db8f11643829560bab55
font@umich.edu	????	1d8d70dddf147d2d92a634817f01b239
collemc@umich.edu	????	acf06cdd9c744f969958e1f085554c8b

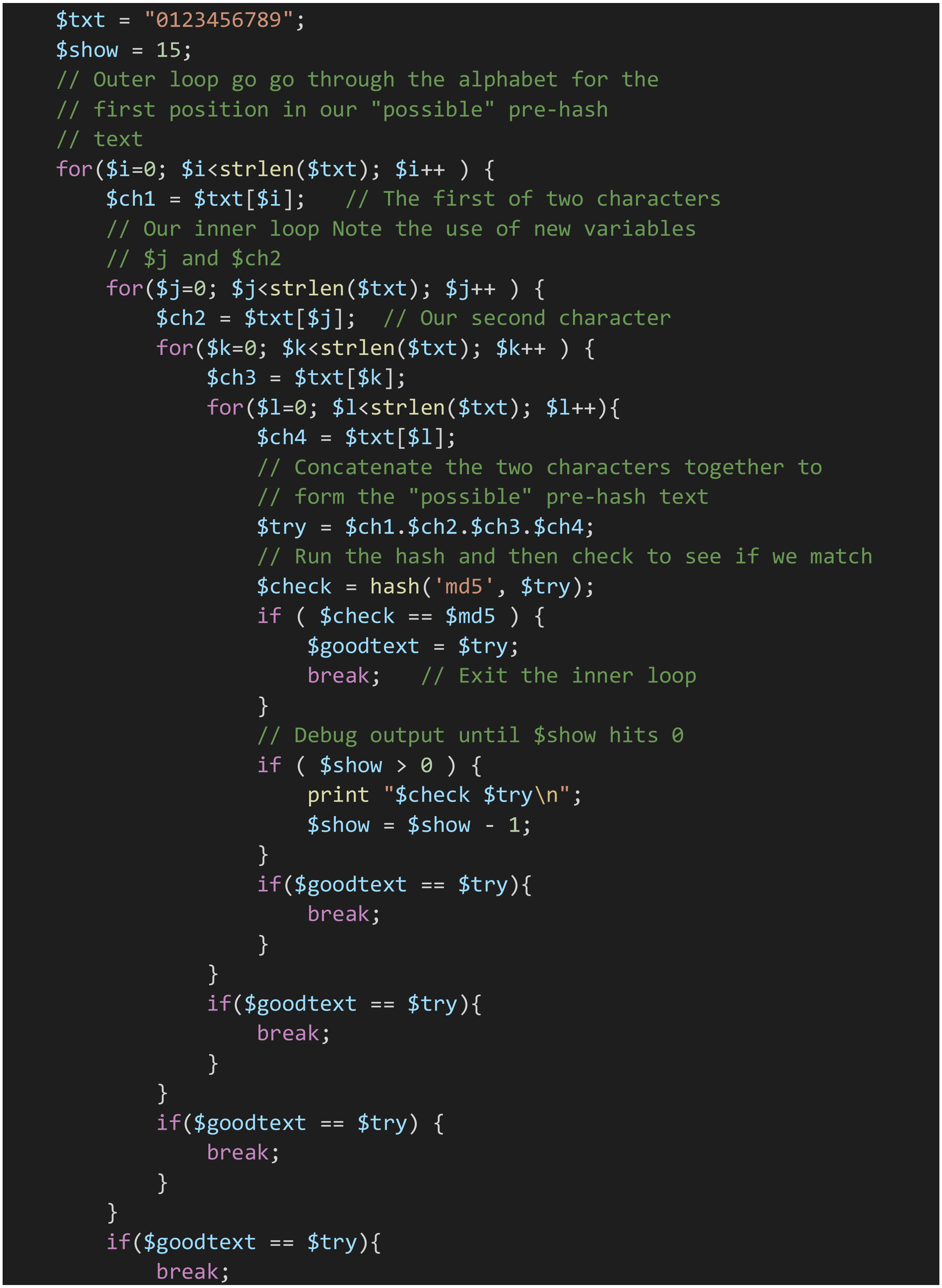
You should be able to easily crack all but one of these these PINs using your application.

The simplest brute force approach generally is done by writing a series of nested loops that go through all possible combinations of characters. This is one of the reasons that password policies specify that you include upper case, lower case, numbers, and punctuation in passwords is to make brute force cracking more difficult. Significantly increasing the length of the password to something like 20-30 characters is a very good to make brute force cracking more difficult.

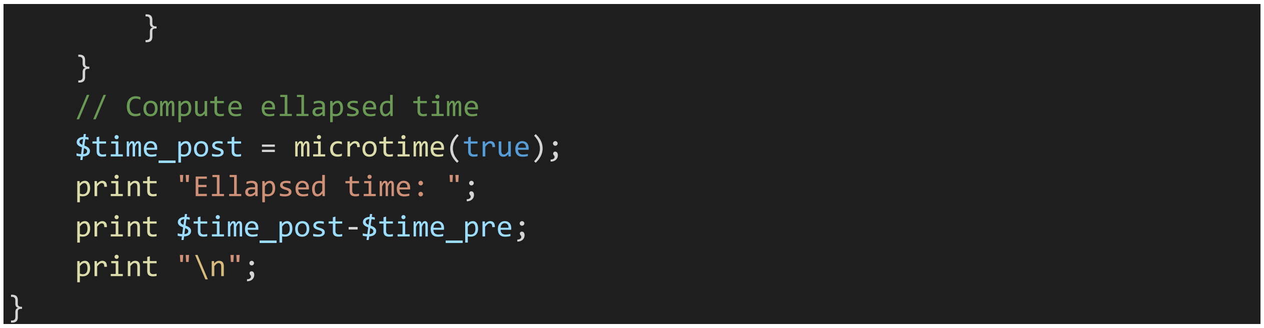
STEP 1: Edit the part of the index.php



```
if ( isset($_GET['md5']) ) {
    $time_pre = microtime(true);
    $md5 = $_GET['md5'];
    // This is our alphabet
```

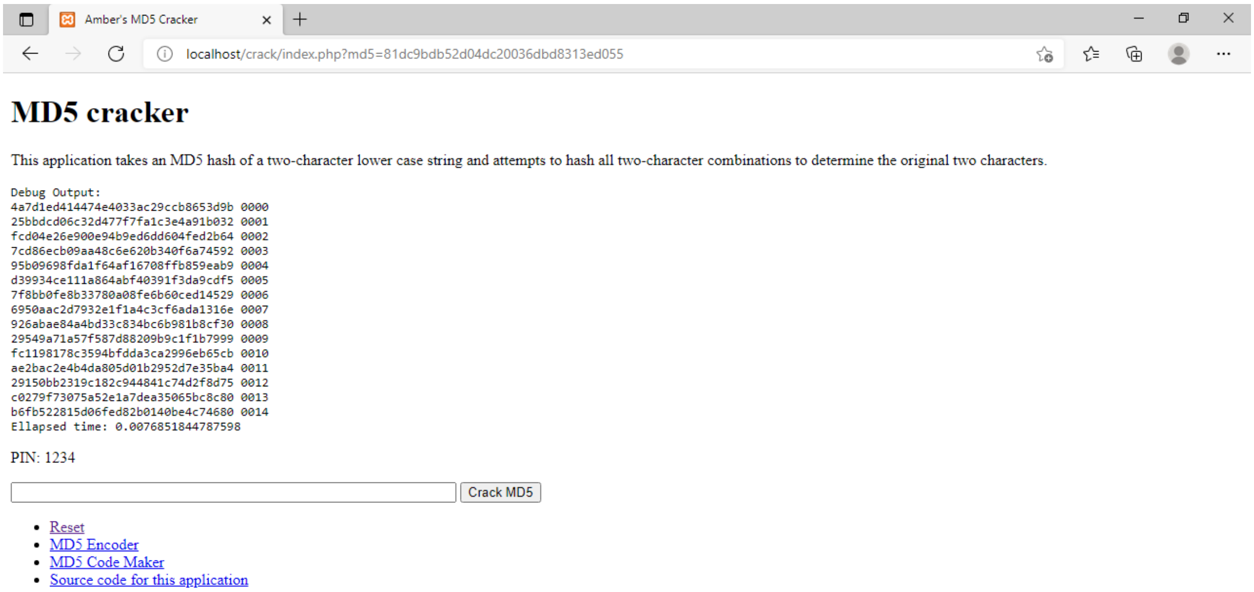


```
    $txt = "0123456789";
    $show = 15;
    // Outer loop go go through the alphabet for the
    // first position in our "possible" pre-hash
    // text
    for($i=0; $i<strlen($txt); $i++ ) {
        $ch1 = $txt[$i];   // The first of two characters
        // Our inner loop Note the use of new variables
        // $j and $ch2
        for($j=0; $j<strlen($txt); $j++ ) {
            $ch2 = $txt[$j];  // Our second character
            for($k=0; $k<strlen($txt); $k++ ) {
                $ch3 = $txt[$k];
                for($l=0; $l<strlen($txt); $l++){
                    $ch4 = $txt[$l];
                    // Concatenate the two characters together to
                    // form the "possible" pre-hash text
                    $try = $ch1.$ch2.$ch3.$ch4;
                    // Run the hash and then check to see if we match
                    $check = hash('md5', $try);
                    if ( $check == $md5 ) {
                        $goodtext = $try;
                        break;   // Exit the inner loop
                    }
                    // Debug output until $show hits 0
                    if ( $show > 0 ) {
                        print "$check $try\n";
                        $show = $show - 1;
                    }
                    if($goodtext == $try){
                        break;
                    }
                }
                if($goodtext == $try){
                    break;
                }
            }
            if($goodtext == $try) {
                break;
            }
        }
        if($goodtext == $try){
            break;
```



```
        }
    }
    // Compute ellapsed time
    $time_post = microtime(true);
    print "Ellapsed time: ";
    print $time_post-$time_pre;
    print "\n";
}
```

STEP 2: Run the Index.php and enter the “81dc9bdb52d04dc20036dbd8313ed055” to get the PIN: 1234


Amber's MD5 Cracker
localhost/crack/index.php?md5=81dc9bdb52d04dc20036dbd8313ed055

MD5 cracker

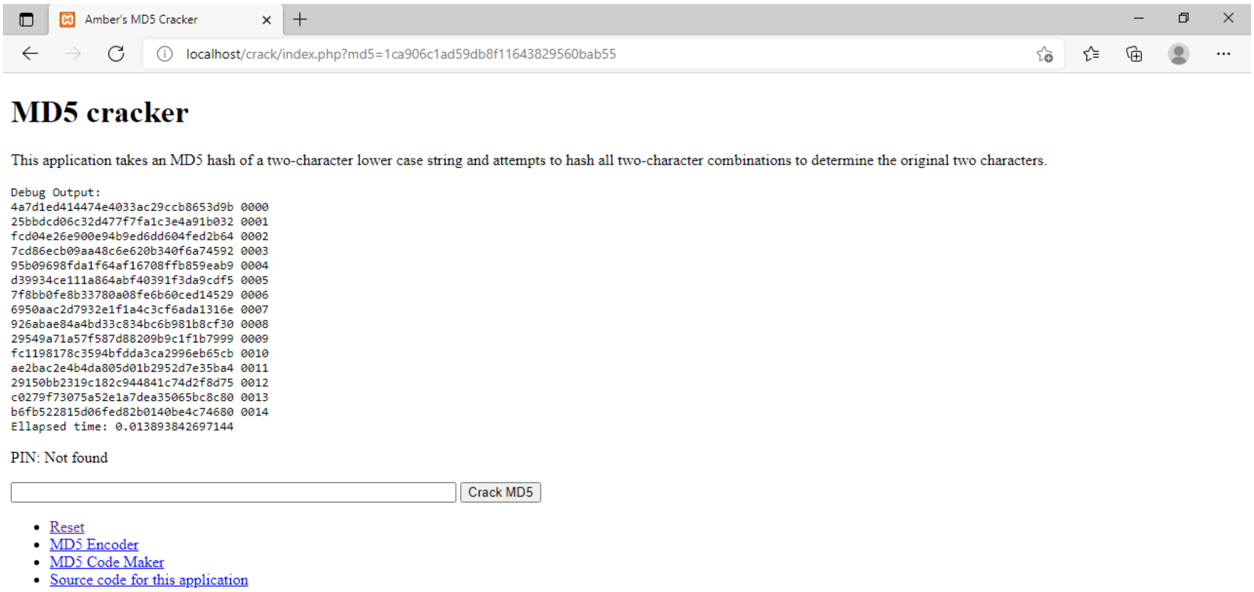
This application takes an MD5 hash of a two-character lower case string and attempts to hash all two-character combinations to determine the original two characters.

```
Debug Output:
4a7d1ed414474e4033ac29ccb8653d9b 0000
25bbdcd06c32d477f7fa1c3e4a91b032 0001
fcd04e26e900e94b9ed6dd604fed2b64 0002
7cd86ecb09aa48c6e620b340f6a74592 0003
95b09698fda1f64af16708ffb859eab9 0004
d39934ce111a864abf40391f3da9cdf5 0005
7f8bb0fe8b33780a08fe6b60ced14529 0006
6950aac2d7932e1f1a4c3cf6ada1316e 0007
926abae84a4bd33c834bc6b981b8cf30 0008
29549a71a57f587d88209b9c1f1b7999 0009
fc1198178c3594bfdda3ca2996eb65cb 0010
ae2bac2e4b4da805d01b2952d7e35ba4 0011
29150bb2319c182c944841c74d2f8d75 0012
c0279f73075a52e1a7dea35065bc8c80 0013
b6fb522815d06fed82b0140be4c74680 0014
Ellapsed time: 0.0076851844787598
```

PIN: 1234

- Reset
- MD5 Encoder
- MD5 Code Maker
- Source code for this application

STEP 3: Enter the PIN “1ca906c1ad59db8f11643829560bab55” to make the PIN: Not Found


Amber's MD5 Cracker
localhost/crack/index.php?md5=1ca906c1ad59db8f11643829560bab55

MD5 cracker

This application takes an MD5 hash of a two-character lower case string and attempts to hash all two-character combinations to determine the original two characters.

```
Debug Output:
4a7d1ed414474e4033ac29ccb8653d9b 0000
25bbdcd06c32d477f7fa1c3e4a91b032 0001
fcd04e26e900e94b9ed6dd604fed2b64 0002
7cd86ecb09aa48c6e620b340f6a74592 0003
95b09698fda1f64af16708ffb859eab9 0004
d39934ce111a864abf40391f3da9cdf5 0005
7f8bb0fe8b33780a08fe6b60ced14529 0006
6950aac2d7932e1f1a4c3cf6ada1316e 0007
926abae84a4bd33c834bc6b981b8cf30 0008
29549a71a57f587d88209b9c1f1b7999 0009
fc1198178c3594bfdda3ca2996eb65cb 0010
ae2bac2e4b4da805d01b2952d7e35ba4 0011
29150bb2319c182c944841c74d2f8d75 0012
c0279f73075a52e1a7dea35065bc8c80 0013
b6fb522815d06fed82b0140be4c74680 0014
Ellapsed time: 0.013893842697144
```

PIN: Not found

- Reset
- MD5 Encoder
- MD5 Code Maker
- Source code for this application

Voila!