

```

<?php
//error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);
//ini_set('display_errors' , 1);
//Receiving from frontend: $data =
array('type'=>'student_answers','username'=>$username,'question_id'=>$id,'questions'=>$questions,
'answers'=>$answers,'test_case_1'=>$test_case_1,'test_case_2'=>$test_case_2,'test_case_3'=>$test_
case_3,'test_case_4'=>$test_case_4,'test_case_5'=>$test_case_5,'question_points'=>$question_point
s);
if($_POST['type']=='student_answers'){
    $problem_array = $_POST['problem'];
    $student_code_array = $_POST['answers'];
    $original_student_code = $student_code_array; //holds all
values of $student_code_array without any fixes
    $question_points = $_POST['question_points']; //The amount of
points a question is worth provided by Professor on make-exam page, PER question

    $test_case_1_array = $_POST['test_case_1'];
    $test_case_2_array = $_POST['test_case_2'];
    $test_case_3_array = $_POST['test_case_3'];
    $test_case_4_array = $_POST['test_case_4'];
    $test_case_5_array = $_POST['test_case_5'];

    $test_cases_array = array($test_case_1_array, $test_case_2_array, $test_case_3_array,
    $test_case_4_array, $test_case_5_array); //Array of arrays

    $test_case_explode_array = []; //Only need one,
can just overwrite values

    $test_case_1_answer = [];
    $test_case_2_answer = [];
    $test_case_3_answer = [];
    $test_case_4_answer = [];
    $test_case_5_answer = [];

    $test_cases_answer_array = array($test_case_1_answer, $test_case_2_answer,
    $test_case_3_answer, $test_case_4_answer, $test_case_5_answer); // Array of Arrays

    $student_function_array = []; //array of
student function name per question; This is the exploded part of the function name + '('
    $student_statement_array = []; //array of
student closing statement ('return' or 'print') per question

    $input_data =
    array('test_cases_array'=>$test_cases_array,'test_cases_answer_array'=>$test_cases_answer_array
    ); //An Associative array of arrays of arrays

    $points = []; //will hold
number of points per question: (# of correct testcases)/(total # of testcases)
    $reduction_function = []; //will hold
number of misspelled function name point reductions per question: (1/total # of
testcases)*25; --> 25 is (1/4) amount of points for a question x 100%;
    $reduction_statement = [];
    $question_grade = []; //will hold the
complete grade PER question: $question_grade = question_points*points

```

```

for($x=0;$x<sizeof($input_data['test_cases_array']);$x++){ //size of
test_cases_array (should be 5 for 5 test cases)
$count=0; //tracks number
of correct test cases PER question
for($y=0;$y<sizeof($input_data['test_cases_array']);$y++){ //size of
test_cases_array (should be 5 for 5 test cases) --> This is necessary so when you do
test_cases_array[$y][$x] it will not stop prematurely, due to $y not incrementing as much
as $x will
if(empty($input_data['test_cases_array'][$y])){ //if there is no
test case values; This will always be the case as long as test cases are created in
numerical order
break; //breaks to the
for loop that will increment $x
}

if(empty($input_data['test_cases_array'][$y][$x])){
if($y != sizeof($input_data['test_cases_array'][$y])-1){ //A check to make
sure that $y doesn't increment if the last test case (test case 5) actually had a 0
value. If so, the count would go out-of-bounds
continue;
}
}
else{
$tmp_id_array = explode(" ",$input_data['test_cases_array'][$y][$x]); //Takes values
separated by commas in string and puts into an array; Every test case will be split
into an _explode_array of size 2, [0]=testcase [1]=intended output
//take out values that are equal to null
$j = 0; //index of
_explode_array
for($i=0; $i<sizeof($tmp_id_array); $i++){ //will separate
empty and null values from array
if($tmp_id_array[$i] != " " && !is_null($tmp_id_array[$i]) &&
preg_match('/^[A-Za-z0-9~!@#%&*()-_+=\\|\/.,<>?]/', $tmp_id_array[$i])){ //Every
test case will be split into an _explode_array of size 2, [0]=testcase [1]=intended
output
test_case_explode_array[$j] = trim($tmp_id_array[$i]);
$j++;
}
}

//save the function name that the student types in and include in an if statement, that
if it is wrong, replace it with the correct one, take points off. If the student
doesn't get it wrong, store the value anyway.
if($y == 0){ //will run for
the first test case for EVERY question
preg_match("/def [A-Za-z0-9]*\(/", $student_code_array[$x],
$student_func_name_with_def); //takes everything in 'def' through '(' in student code
array; $student_func_name_with_def[0]='def' + function name student typed + '('
preg_match("/[A-Za-z0-9]*\(/", $student_func_name_with_def[0],
$student_func_name); //takes out the 'def' part of $student_func_name_with_def
and only leaves the function name including '('; $student_func_name[0]= function name
student typed + '('

```

```

preg_match("/[A-Za-z0-9]*\(/", $test_case_explode_array[0],
$correct_func_name);          //takes first part of the test case function name
including '(' inputted from $test_case_explode_array[0] and places it into array
$correct_func_name

$student_function_array[$x] = $student_func_name[0];          //stores
student's function name + '(' per question

if($student_func_name[0] != $correct_func_name[0]){
    $student_code_array[$x] = preg_replace("/def [A-Za-z0-9(,)]*\(/", 'def
    '.$correct_func_name[0], $student_code_array[$x]); //replaces everything in 'def'
    through '('; $correct_func_name[0]=what to replace; $student_code_array[$x]=input
    string to search in
    $total_test_cases = test_case_total($x, $input_data['test_cases_array']);
    $reduction_function[$x] = (1/$total_test_cases)*25;          //(1/total # of
    testcases)*25; --> 25 is (1/4) amount of points for a question x 100% point
    reduction for misspelled function name.
}
else{
    $reduction_function[$x] = 0;
}

preg_match("/return|print/", $problem_array[$x], $correct_statement);          //checks
question for 'return' or 'print' and stores into $correct_statement array
preg_match("/return|print/", $student_code_array[$x], $student_statement);          //checks
student's code for 'return' or 'print' and stores into $student_statement array
$student_statement_array[$x] = $student_statement[0];          //stores
student's closing statement ('return' or 'print') per question

if($student_statement[0] != $correct_statement[0] && $correct_statement[0] == 'print'){
    $total_test_cases = test_case_total($x, $input_data['test_cases_array']);
    $reduction_statement[$x] = (1/$total_test_cases)*25;          //(1/total # of
    testcases)*25; --> 25 is (1/4) amount of points for a question x 100% point
    reduction for misspelled function name.
    $student_code_array[$x] = preg_replace("/return/", "print", $student_code_array[$x]);
}
elseif($student_statement[0] != $correct_statement[0] && $correct_statement[0] ==
'return'){
    $total_test_cases = test_case_total($x, $input_data['test_cases_array']);
    $reduction_statement[$x] = (1/$total_test_cases)*25;          //(1/total # of
    testcases)*25; --> 25 is (1/4) amount of points for a question x 100% point
    reduction for misspelled function name.
    $student_code_array[$x] = preg_replace("/print/", "return", $student_code_array[$x]);
}
else{
    $reduction_statement[$x] = 0;
}
}

if(!empty($test_case_explode_array)){
    $overwrite_execution = shell_exec("python
    /afs/cad.njit.edu/u/r/l/rl265/public_html/php/overwrite.py '$student_code_array[$x]'
    '$test_case_explode_array[0]'");          //runs "overwrite.py" file with 2 arguments;

```

```

$student_code[$x]=student's code for 1 question; $test_case_explode_array[0]=testcase
$overwrite_answer = shell_exec("python
/afs/cad.njit.edu/u/r/l/rl265/public_html/php/student_answer.py"); //runs
"student_answer.py" file and stores output into "$overwrite_answer"
}

if(trim($overwrite_answer) == trim($test_case_explode_array[1]) &&
preg_match('/^[A-Za-z0-9~!@#%&*()-_+=\\|\/.,<>?]/',
trim($overwrite_answer))) { //if the python output == test case solution
    $input_data['test_cases_answer_array'][$y][$x] = '1'; //test case
    answer for each test case in each question (remember 1 question can have up to 5 test
    cases each)
    $count = $count + 1;
}
else{
    $input_data['test_cases_answer_array'][$y][$x] = '0';
}

$total_test_cases = test_case_total($x, $input_data['test_cases_array']);
$points[$x] = ($count)/($total_test_cases); //(# of correct
testcases)/(total # of testcases) --> Points Scaling Factor
$question_grade[$x] = $question_points[$x]*$points[$x] - ($reduction_function[$x] +
$reduction_statement[$x]); //Complete grade PER question: question_points*points -
($reduction_function[$x] + $reduction_statement[$x])
}
unset($overwrite_answer); //prevents
variable from storing previous iterations values
unset($test_case_explode_array); //resets variable
to null
}
unset($student_func_name_with_def);
unset($student_func_name);
unset($correct_func_name);
unset($student_statement);
unset($correct_statement);
}

//get exam grade
$exam_grade = 0;
for($i=0; $i<sizeof($question_grade); $i++){
    $exam_grade += $question_grade[$i];
}

$data = array
('type'=>'student_answers','username'=>$_POST['username'],'question_id'=>$_POST['question_id'],
'answers'=>$student_code_array,'original_student_code'=>$original_student_code,'test_case_1_ans
wer'=>$input_data['test_cases_answer_array'][0],'test_case_2_answer'=>$input_data['test_cases_a
nswer_array'][1],'test_case_3_answer'=>$input_data['test_cases_answer_array'][2],'test_case_4_a
nswer'=>$input_data['test_cases_answer_array'][3],'test_case_5_answer'=>$input_data['test_cases
_answer_array'][4],'points'=>$points,'reduction_function'=>$reduction_function,'reduction_state
ment'=>$reduction_statement,'student_function_array'=>$student_function_array,'student_statemen
t_array'=>$student_statement_array,'question_grade'=>$question_grade,'exam_grade'=>$exam_grade)
;

```

```

$string = http_build_query($data);
$ch = curl_init("https://web.njit.edu/~rl265/php/backend.php");
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_exec($ch);
curl_close($ch);

}

function test_case_total($x, $test_cases_array){ //passing the
following: $x=the value of the question on; $test_cases_array=$input_data['test_cases_array'];
$total_test_cases = 0; //total number of
test cases PER question
    for($i=0; $i<sizeof($test_cases_array); $i++){ //size of number
of test cases
        if(sizeof($test_cases_array[$i][$x])==0){ //To avoid 'Index
Out Of Bounds' Error. (size of the value of test_case_#_array) --> Takes first value of
every test_case_# array (so takes all test cases[1-5] PER question)
            $i++;
        }
        else{
            if(preg_match('/^[A-Za-z0-9`~!@#%&*()-_+=\\|\/.,<>?]/', $test_cases_array[$i][$x])){
                $total_test_cases++;
            }
        }
    }
    return $total_test_cases;
}
?>

```