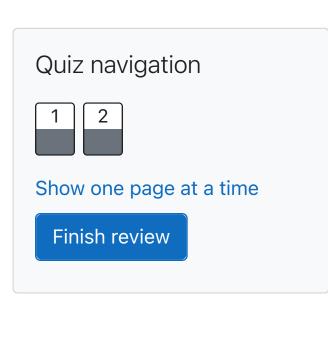
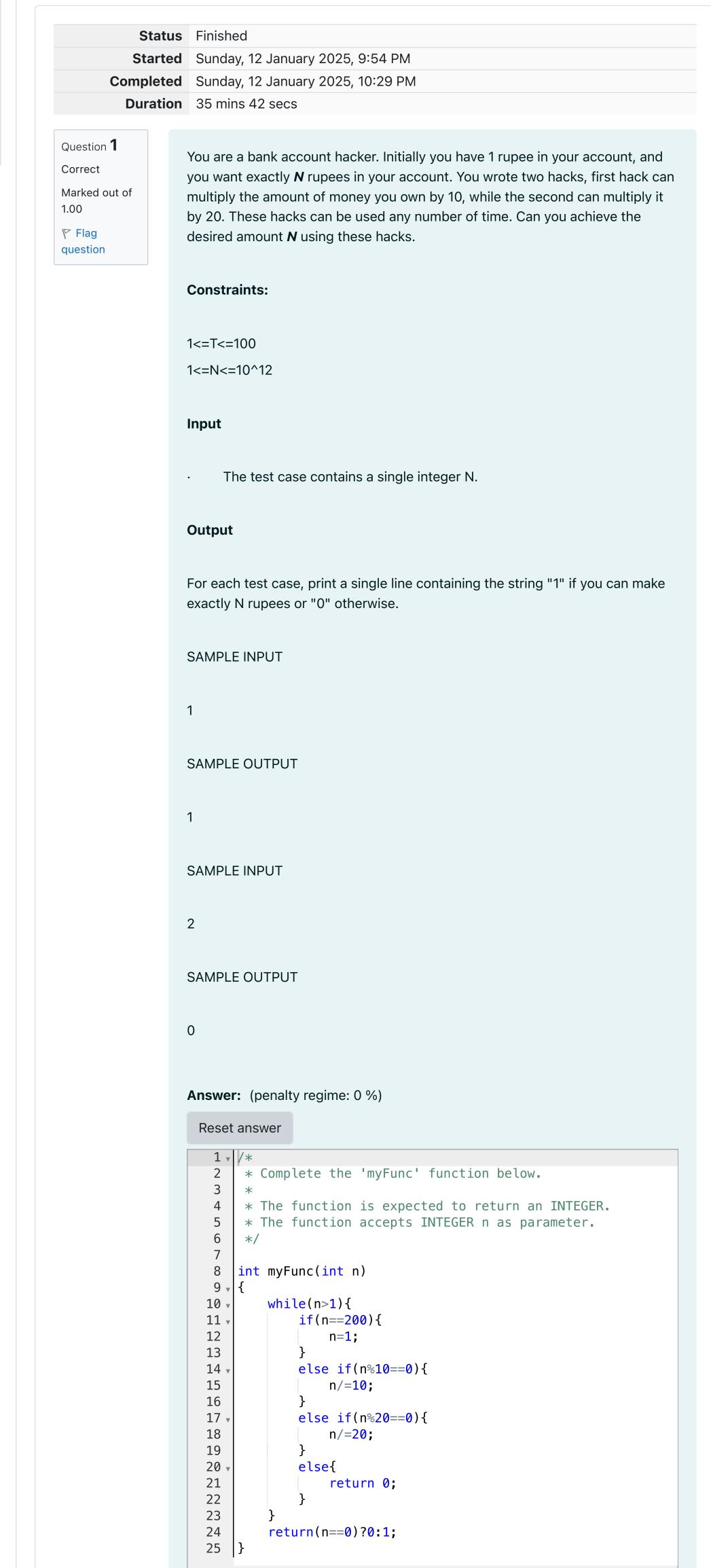
GE23131-Programming Using C-2024





printf("%d", myFunc(10)) 1 1 **/** printf("%d", myFunc(25)) 0 printf("%d", myFunc(200)) 1 1 **/** Passed all tests! < Question **2** Find the number of ways that a given integer, X, can be expressed as the sum Correct of the N^{th} powers of unique, natural numbers. Marked out of 1.00 For example, if X = 13 and N = 2, we have to find all combinations of unique ▼ Flag squares adding up to 13. The only solution is $2^2 + 3^2$. question **Function Description** Complete the powerSum function in the editor below. It should return an integer that represents the number of possible combinations. powerSum has the following parameter(s): X: the integer to sum to N: the integer power to raise numbers to Input Format The first line contains an integer **X**. The second line contains an integer **N**. **Constraints** $1 \le X \le 1000$ $2 \le N \le 10$ **Output Format** Output a single integer, the number of possible combinations calculated. Sample Input 0 10 2 **Sample Output 0 Explanation 0** If X = 10 and N = 2, we need to find the number of ways that 10 can be represented as the sum of squares of unique numbers.

 $10 = 1^2 + 3^2$

Expected Got

1

1

0

/

✓

Test

printf("%d", myFunc(1))

printf("%d", myFunc(2))

This is the only way in which 10 can be expressed as the sum of unique squares. **Sample Input 1** 100 2 **Sample Output 1** 3 **Explanation 1** $100 = (10^2) = (6^2 + 8^2) = (1^2 + 3^2 + 4^2 + 5^2 + 7^2)$ **Sample Input 2** 100 3 Sample Output 2 **Explanation 2** 100 can be expressed as the sum of the cubes of 1, 2, 3, 4. (1 + 8 + 27 + 64 = 100). There is no other way to express 100 as the sum of cubes. **Answer:** (penalty regime: 0 %) Reset answer 1 | /* * Complete the 'powerSum' function below. 3 * The function is expected to return an INTEGER. 4 * The function accepts following parameters: * 1. INTEGER x 7 * 2. INTEGER n 8 */ 9 int powerSum(int x, int m, int n) 10 11 ▼ { **if**(x==0){ 12 • 13 return 1; 14 if(x<0){</pre> **15** ▼ return 0; 16 17 18 int count =0; 19 • for(int i=m; ;i++){ 20 int power=1; 21 • for(int j=0;j<n;j++){</pre> 22 power*=i; 23 if(power>x){ 24 • 25 break; 26 27 count+=powerSum(x-power,i+1,n); 28 29 return count; 30 31 } **Expected Got** Test printf("%d", powerSum(10, 1, 2)) | 1 1 **/** Passed all tests! < Finish review