## Gradescope

1. Nosotros les proporcionaremos un código base de donde deberán partir para completar dicho ejercicio. Este archivo es llamado solution.py y lo encontrarán en la indicación de la tarea en CANVAS.
2. Al finalizar, **solo** subir el archivo solution.py (NO cambiar el nombre del archivo y NO comprimirlo).

1

1. Cada pregunta tiene diversos casos de prueba. Para obtener la nota completa en una pregunta, el algoritmo debe obtener la respuesta correcta en dichos casos de prueba.
2. Si un caso de prueba falla, visualizarán un mensaje de error con sugerencias. **Lee el error**, revisa el código e inténtalo de nuevo.
3. Los input de los casos de prueba son confidenciales.

## Indicaciones específicas

1. En el anexo 1, se puede ver la plantilla de código.
2. Ustedes deben escribir dentro de la sección y a la misma altura de donde esta escrito *"Código comienza aquí"*. Además, no deben modificar nada debajo de *"Código acaba aquí"*. Recuerden tener cuidado con las indentaciones.
3. Los input del ejercicio se encuentran en la plantilla. Recuerden usar estas variables para resolver el ejercicio.
4. La respuesta del ejercicio debe ser impresa **especificamente** con *print()* para que Gradescope la tome en consideración. En el caso de funciones, esta debe ser retornada, según especifique la plantilla otorgada.
5. Al momento de la impresión de la respuesta, no adicionar texto. Imprimir **únicamente** el resultado que pide el ejercicio. De la misma manera con las funciones.

## Caso: Ataque de los Titantes - (20 pts)

Tu labor en esta ocasión es ayudar a Eren Jaeger. Para ello, debes determinar por él si ha aprobado el Programa Formativo como soldado o no para poder formar parte de Legión de Reconocimiento. La única forma de aprobar dicho programa es aprobando todos y cada uno de los *N* cursos que lo componen.

* Cada curso está compuesto por *Mi* notas donde i va de 1 a N.
* No necesariamente todos los cursos tienen la misma cantidad de notas.
* Cada curso está identificado por un código único diferente.
  + En caso el código del curso sea par, será necesario que el promedio de las notas de dicho curso sea mayor o igual a 11.
  + En caso el código del curso sea impar, la única forma de aprobar es obtener al menos una nota que sea mayor o igual a 16.

**Parámetros del programa**

1. El numero entero *N* que representa la cantidad de cursos.
2. Una lista *codigos* de longitud N que representa los códigos de los cursos.
3. Una lista *M* de longitud N donde cada elemento *Mi* representa la cantidad de notas del curso con el código en la misma posición *i*.
4. Una matriz *notas* de tamaño *N* · *Mi*. Que representa todas las notas de Eren.
5. Recuerde que usted debe usar los parámetros de las funciones, no debe adicionar inputs.

**Entrega de resultados**

1. Si N es un número menor o igual a 0. Usted deberá retornar “Código de autodestrucción detectado. Iniciando proceso en 3, 2, 1”. Y no ejecutará la lógica del cálculo de notas.
2. Si cualquier *Mi* cantidad de notas es un número menor o igual a 0. Usted deberá retornar “Error, no pueden existir cursos sin notas”. Y no ejecutará la lógica del cálculo de notas.
3. En el caso de aprobar el Programa Formativo como soldado, usted debe retornar “Felicitaciones. Te graduaste”.
4. De lo contrario, debe retornar ’Has reprobado. Pero no te rindas, toma el Programa nuevamente. Aprobaste *X* de los *N* cursos’. Donde X es la cantidad de cursos aprobados por Eren y N es el total de cursos que fue indicado al inicio por Eren.
5. En caso se inserten 2 códigos de curso iguales, se debe retornar ‘Error, ha introducido el mismo curso 2 veces’. Y no ejecutará la lógica del cálculo de notas.

Usted solo debe retornar un único mensaje de error, en el caso de que usted detecte múltiples errores diferentes al mismo tiempo, su programa debe retornar el error con menor índice según la numeración presentada anteriormente (números más pequeños simboliza mayor prioridad). Por ejemplo, si usted detecta error de la regla 1, pero a la vez el error de la regla 5. Solo debe retornar el mensaje ‘Código de autodestrucción detectado. Iniciando proceso en 3, 2, 1’. **Tenga en cuenta que las únicas reglas consideradas como error son 1,2 y 5**.

**Ejemplo 1.**

***Parámetros*** :

|  |
| --- |
| N= 3  codigos=[10001,10002,99991] M=[2,3,1]  notas=[[16,9],[1,20,15],[20]] |

1

2

3

4

***Retorno*** :

|  |
| --- |
| Felicitaciones. Te graduaste |

1

***Explicación*** Se debe evaluar curso por curso si se cumple la condición de acuerdo al tipo de código de curso que este tenga. Para el curso ‘10001’ se debe cumplir que tenga al menos una nota mayor o igual a 16, lo cual es Verdadero. De la misma manera ocurre con ‘99991’. En el caso de ‘10002’ se sacó el promedio y se verificó que cumple la condición propuesta. Por lo que, al haber aprobado todos los cursos, Eren ha logrado graduarse del Programa de Formación de soldados.

**Ejemplo 2.**

***Parámetros*** :

|  |
| --- |
| N= 2  codigos=[10001,10002] M=[2,3] notas=[[15,9],[1,20,15]] |

1

2

3

4

***Retorno*** :

|  |
| --- |
| Has reprobado. Pero no te rindas, toma el Programa nuevamente. Aprobaste  1 de los 2 cursos |

1

***Explicación*** El curso con código 10001 no cumple con la condición de tener al menos una nota igual o mayor a 16. Por ende, al no haber aprobado todos los cursos, Eren ha reprobado el Programa formativo.

**Ejemplo 3.**

***Parámetros*** :

|  |
| --- |
| N=0 codigos=[] M=[] notas=[] |

1

2

3

4

***Retorno*** :

|  |
| --- |
| Código de autodestrucción detectado. Iniciando proceso en 3, 2, 1 |

1

***Explicación*** El valor de N es menor o igual a 0, por ello, se activó el código de autodestrucción.

**Ejemplo 4.**

***Parámetros*** :

|  |
| --- |
| N= 3  codigos=[12345,99981,12331] M=[2,-1,2] notas=[[20,19],[],[10,11]] |

1

2

3

4

***Retorno*** :

|  |
| --- |
| Error, no pueden existir cursos sin notas |

1

***Explicación*** El valor de algún *Mi* es menor o igual a 0, por ello, se detectó un error (curso sin notas).

**Ejemplo 5.**

***Parámetros*** :

|  |
| --- |
| N= 3  codigos=[12345,12345,12331] M=[2,1,2]  notas=[[20,19],[1],[10,11]] |

1

2

3

4

***Retorno*** :

|  |
| --- |
| Error, ha introducido el mismo curso 2 veces |

1

***Explicación*** se insertó dos veces un curso con código 12345. Por ello se imprime el mensaje de error.

## 1. Anexos

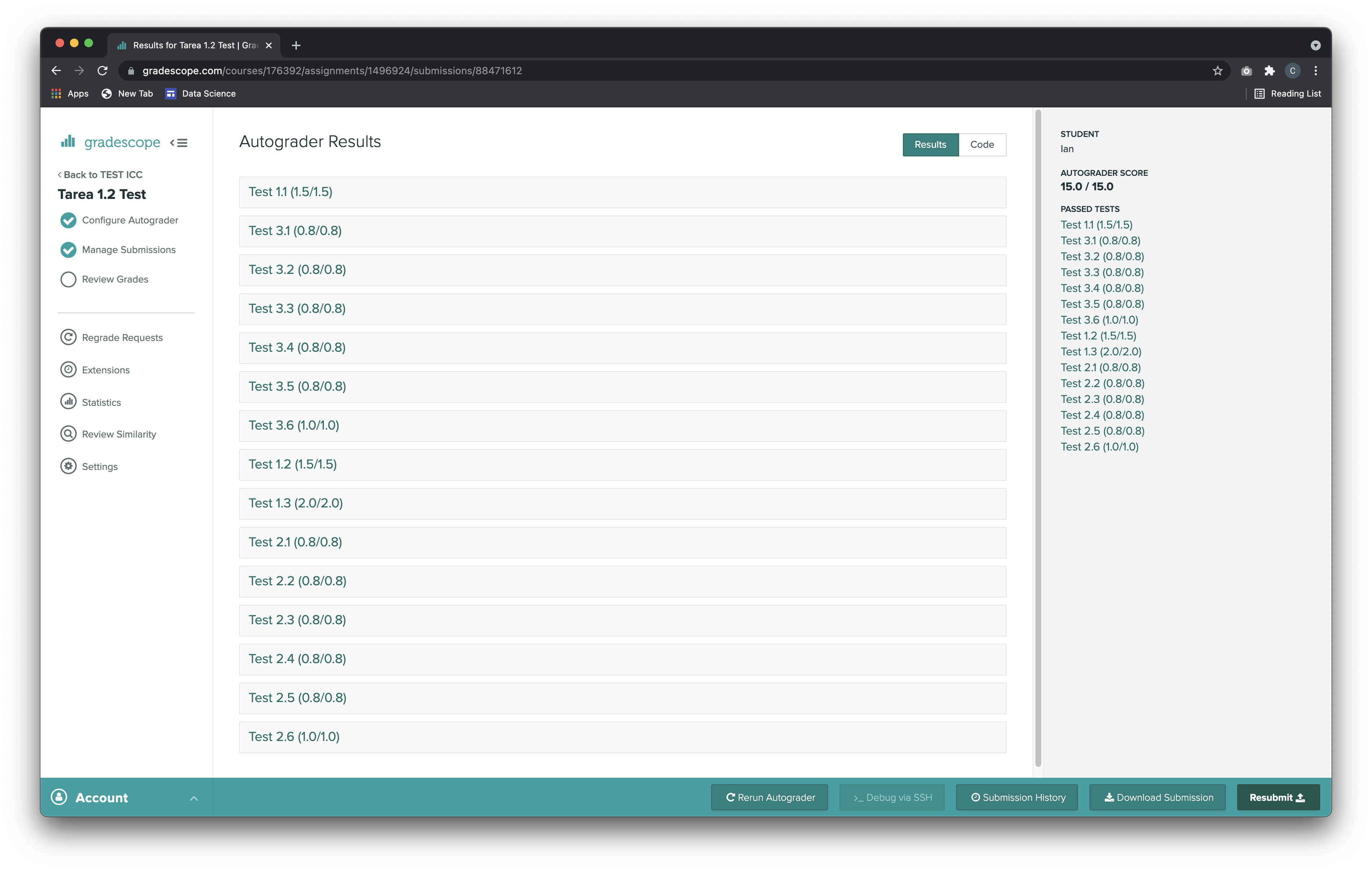


Figure 1: Casos de prueba correctos en Gradescope.

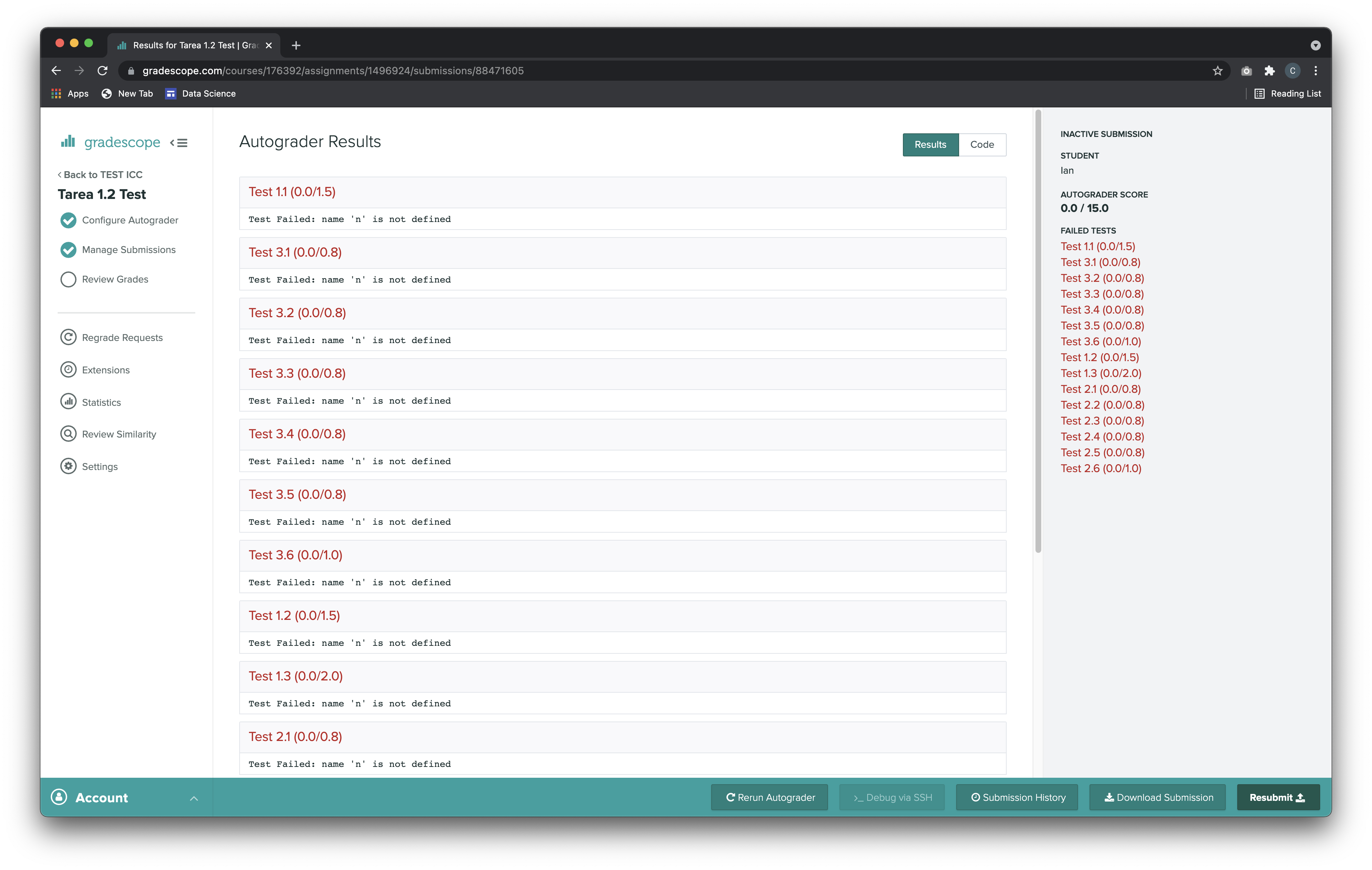


Figure 2: Entrega incorrecta en Gradescope.

|  |  |
| --- | --- |
| class | Solution(): |
| def | casoDesarrollo(self, N, codigos=[], M=[], notas=[]):  resultado=""  #SU SOLUCION EMPIEZA AQUI  #SU SOLUCION TERMINA AQUI return resultado |

1

2

3

4

5

6

7

8

9

10

11

12

13

Listing 1: Template solution.py.