HENNGE

# CHALLENGE DETAILS

## Introduction

Thank you for your application to HENNGE.

This challenge is intended for **backend position or global internship** applicants.

This challenge consists of 3 missions. After successful completion of Mission 3, you will receive an email with a link to submit your CV and Cover Letter (CL). Once we receive them, our engineers will review your code and documents (CV & CL) altogether, and we will contact you via email with the screening result. If you haven't received your result within a month, please let us know!

**Tips:** Due to the competitiveness of our application process, we highly recommend applicants to explain their motivation clearly in the Cover Letter (CL).

## Mission 1: Write a program which fulfills the requirements below

### Description

- We want you to calculate the sum of squares of given integers, excluding any negatives.
- The first line of the input will be an integer `N` ( `1 <= N <= 100` ), indicating the number of test cases to follow.
- Each of the test cases will consist of a line with an integer `X` ( `0 < X <= 100` ), followed by another line consisting of `X` number of space-separated integers `Yn` ( `-100 <= Yn <= 100` ).
- For each test case, calculate the sum of squares of the integers, excluding any negatives, and print the calculated sum in the output.
- **Note:** There should be no output until all the input has been received.
- **Note 2:** Do not put blank lines between test cases solutions.
- **Note 3:** Take input from standard input, and output to standard output.

### Rules

Write your solution using Go Programming Language or Python Programming Language. **Do not** submit your solution with both languages at once!

You may only use standard library packages. In addition, extra point is awarded if solution does not declare any global variables.

### Specific rules for Go solution

- Your source code must be a single file
- Do not use any `for` and `goto` statement
- Your solution will be tested against Go 1.20 (as of February 2023) or higher

```
package main

func main() {
  ...
}
```

### Specific rules for Python solution

- Your source code must be a single file, containing at least a main function
- Do not use any `for` loop, `while` loop, or any list / set / dictionary comprehension
- Your solution will be tested against Python 3.11 (as of February 2023) or higher

```
def main():
    ...


if __name__ == "__main__":
    main()
```

### Sample Input

```
2
4
3 -1 1 14
5
9 6 -53 32 16
```

### Sample Output

```
206
1397
```

# Mission 2: Publish your source code as a secret gist

## Description

Publish your source code as a secret gist. You will need a GitHub account to create a secret gist, if you are not familiar with *secret gists*, follow the instructions here.

Please make sure **not** to publish it as a public gist.

We will take a look at your source code later, after you achieve Mission 3.

Your program will be **auto-tested**, so please be **strict** about the input/output specification. For failed solution, we will still assess it case-by-case but it will cause points reduction to your overall score.

**Note:** Make sure to upload only a single file as your solution and no additional files should exist in your provided gist.

# Mission 3: Send us the URL of your work

## Description

First, construct a JSON string like below:

#### Go solution

```
{
  "github_url": "https://gist.github.com/YOUR_ACCOUNT/GIST_ID",
  "contact_email": "EMAIL",
  "solution_language": "golang"
}
```

#### Python solution

```
{
  "github_url": "https://gist.github.com/YOUR_ACCOUNT/GIST_ID",
  "contact_email": "EMAIL",
  "solution_language": "python"
}
```

Fill in your email address for `EMAIL`, and the path to your secret gist for `YOUR_ACCOUNT/GIST_ID`. In addition, `solution_language` should be filled with the solution language of your choice (`golang` or `python`). Be sure you have double-checked your email address; we will contact you by email.

Then, make an HTTP POST request to the following URL with the JSON string as the body part.

```
https://api.challenge.hennge.com/challenges/003
```

#### Content type

The `Content-Type:` of the request must be `application/json`.

### Authorization

The URL is protected by HTTP Basic Authentication, which is explained on Chapter 2 of RFC2617, so you have to provide an `Authorization:` header field in your POST request

- For the `userid` of HTTP Basic Authentication, use the same email address you put in the JSON string.
- For the `password`, provide a 10-digit time-based one time password conforming to RFC6238 TOTP.

#### Authorization password

For generating the TOTP password, you will need to use the following setup:

- You have to read RFC6238 (and the errata too!) and get a correct one time password by yourself.
- TOTP's `Time Step X` is 30 seconds. `T0` is 0.
- Use `HMAC-SHA-512` for the hash function, instead of the default `HMAC-SHA-1`.
- Token shared secret is the userid followed by ASCII string value "`HENNGECHALLENGE003`" (not including double quotations).

#### Shared secret examples

- For example, if the userid is *"ninja@example.com"*, the token shared secret is *"ninja@example.comHENNGECHALLENGE003"*.
- For example, if the userid is *"ninjasamuraisumotorishogun@example.com"*, the token shared secret is *"ninjasamuraisumotorishogun@example.comHENNGECHALLENGE003"*

If your POST request succeeds, the server returns HTTP status code `200`.

## Rules

- You do not have to disclose how you achieved this mission at this time. Do not hesitate to use source codes or tools on the net, but do the exploring process by yourself of course, do not ask your friend to help you. The only thing that matters is that it works!
- No bruteforce attacks, please!

#### Sample Request

```
POST /challenges/003 HTTP/1.1
Authorization: Basic bmluamFAZXhhbXBsZS5jb206MTU5NTk0MjU2MA==
Host: api.challenge.hennge.com
Accept: */*
Content-Type: application/json
Content-Length: 133

{"github_url":"https://gist.github.com/hennge/b859bd12e7a7fb418141","contact_email":"
```

### Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Wed, 26 Jun 2019 02:15:16 GMT
Transfer-Encoding: chunked

{"message":"Congratulations! You have achieved mission 3"}
```

## Any questions? Found a bug?

Quick reminder: no deadline. Please note that our team will not answer to any inquiry related to the challenge itself. If you have any (other) questions, please reach us through `gip@hennge.com` (for the global internship program) or `recruit-engineer@hennge.com` (for the backend position).

---