

Thử thách trong chương 1

Chương trình 1 - BasicStatistics

Viết chương trình nhận danh sách các số (hợp lệ) từ tham số dòng lệnh (command-line argument), thực hiện tính toán và hiển thị ra màn hình các thông tin sau:

- Số nhỏ nhất, số lớn nhất (min, max)
- Giá trị trung bình (mean)
- Giá trị trung vị (median)

Ghi chú:

- Median gọi là trung vị. Đây chính là giá trị của phần tử ở chính giữa một dãy giá trị có xếp theo thứ tự. Trong trường hợp dãy có số phần tử là chẵn thì trung vị được tính là trung bình của 2 phần tử ở giữa của dãy có thứ tự. (<https://ThachLN.github.io>)
- Giả định dữ liệu truyền trên tham số dòng lệnh là hợp lệ, không cần viết mã kiểm tra.

Chương trình 2 - ReadNumber

Viết chương trình nhận một số nguyên từ tham số dòng lệnh. Số này có tối đa 3 chữ số. Chương trình hiển thị dạng văn bản của số đó. Ví dụ:

123 → “Một trăm hai mươi ba.”

(Văn bản không bao gồm cặp dấu nháy đôi)

Ghi chú:

- Giả định dữ liệu truyền trên tham số dòng lệnh là hợp lệ, không cần viết mã kiểm tra.
- Khuyến khích bạn chủ động nâng cấp chương trình để hỗ trợ số có tối đa 6, 9, 12 chữ số.

Chương trình 3 – Tạo dữ liệu mẫu

Viết chương trình sinh ra bảng dữ liệu ngẫu nhiên từ các tham số dòng lệnh có dạng như sau:

```
Số_dòng  tên_cột_1  kiểu_dữ_liệu_1  tên_cột_2  kiểu_dữ_liệu_2 ...
```

Diễn giải:

- Số_dòng: là một số nguyên cho biết số dòng dữ liệu cần tạo ra
- Cột dữ liệu thứ **nhất** có tên là **tên_cột_1** và có kiểu dữ liệu như **kiểu_dữ_liệu_1**.
- Cột dữ liệu thứ **hai** có tên là **tên_cột_2** và có kiểu dữ liệu như **kiểu_dữ_liệu_2**.
- Tương tự cột dữ liệu thứ 3, thứ 4...thứ n nếu có các cặp tham số tiếp theo,

Ghi chú:

- Giả định dữ liệu truyền trên tham số dòng lệnh là hợp lệ, không cần viết mã kiểm tra.

Chương trình 4 – Cộng 2 số

Viết chương trình thực hiện cộng 2 số với cách sử dụng bằng cách gõ lệnh như sau:

```
sum 235 67
```

Kết quả hiển thị:

Các bước thực hiện:

Lấy '5' cộng '7' => Kết quả được 12, lưu 2, nhớ 1.

Lấy '3' cộng '6' => Kết quả được 6, cộng với nhớ 1 được 10, lưu 0, nhớ 1.

Lấy '2' cộng ' ' => Kết quả được 2, cộng với nhớ 1 được 3, lưu 3.

Kết quả cuối cùng: 312

Diễn giải yêu cầu:

- Yêu cầu viết hàm để thực hiện chức năng chính có dạng là tham số kiểu chuỗi như sau: `sum(st1 string, st2 string) string`

Khi nhận tham số từ dòng lệnh vào thì mặc định là chuỗi nên việc gọi hàm này rất thuận tiện.

- Phiên bản đầu tiên được PUSH lên GITLAB.COM chưa cần xử lý các trường hợp dữ liệu sai.

- Khuyến khích nâng cấp tiếp chương trình trên GITLAB.COM để xử lý các trường hợp dữ liệu sai và hoàn thiện phần mềm `sum`. Không cần thực hiện kiểm tra lỗi đầu vào, mà nên xử lý khi thực hiện tính toán: khi gặp ký tự không phải là ký số (digit) thì dừng và báo lỗi luôn.

Ghi chú:

- Đóng gói và phát hành (Release) chương trình `sum` lên Internet, kèm hướng dẫn để người dùng tải về, lưu vào thư mục mà họ muốn (ví dụ: `D:\RunNow\`). Sau đó người dùng có thể mở cửa sổ lệnh của hệ điều hành, đứng tại bất kỳ đường dẫn nào cũng có thể chạy được lệnh `sum`.
 - o Nếu gõ lệnh `sum` không có tham số thì hiển thị hướng dẫn.
 - o Nếu quá trình tính toán gặp dữ liệu không hợp lệ thì báo lỗi.

Chương trình 5 – Đếm ngày

Viết chương trình `CountDate` để tính tổng số ngày giữa 2 ngày được truyền từ tham số dòng lệnh. Ví dụ gõ lệnh như sau:

```
CountDate 2019-12-31 2020-12-31
```

Có mục đích là đếm số ngày từ ngày 31 tháng 12 năm 2019 đến ngày 31 tháng 12 năm 2020.

Trường hợp không gõ tham số thứ hai như:

```
CountDate 2019-12-31
```

thì có mục đích là đếm số ngày từ ngày 31 tháng 12 năm 2019 đến ngày hiện tại của máy tính.

Tạm thời không cần xử lý các trường hợp ngoại lệ.

Thử thách cho chương 2

Chương trình `ReadCSV_EmployeeV1`

Viết chương trình đọc file csv với đường dẫn được truyền trên tham số dòng lệnh. File csv gồm có các cột thông tin:

- Mã nhân viên: kiểu chuỗi 10 ký tự có định dạng `ABC#####`
 - o A: ký tự từ 'A' tới 'Z' cho biết ngạch nhân viên (C: điều hành; M: Quản lý; E: chuyên gia, N: nhân viên bình thường)

- B: kí tự từ ‘0’ tới ‘9’ cho biết cấp bậc (level) của nhân viên trong ngạch.
- C: kí tự từ ‘A’ tới ‘Z’ cho biết phòng ban hoặc chuyên môn chính của nhân viên (D: Ban giám đốc; K: Kế toán; S: Kinh doanh;)
- Tên (First Name): kiểu chuỗi
- Chữ lót (Middle Name): kiểu chuỗi
- Họ (Last Name): kiểu chuỗi
- Ngày tháng năm sinh: định dạng yyyy-mm-dd
- Hệ số năng lực: số từ 1 đến 9
- Hệ số lương: số thực
- Giới tính: chữ “Nam” hoặc “Nữ” hoặc “Khác”

Chương trình thực hiện các chức năng sau:

- Hiển thị ra màn hình các thông tin:
 - Tổng số nhân viên
 - Tổng số nhân viên theo giới tính:
 - Nam: ?
 - Nữ: ?
 - Khác ?
- Độ tuổi trung bình của toàn bộ nhân viên trong công ty (Năm hiện tại lấy từ thời gian của máy tính đang chạy chương trình).

Gợi ý:

- Sử dụng cấu trúc (struct) để mô tả dữ liệu của nhân viên.

Chương trình ReadCSV_EmployeeV2

Nâng cấp chương trình ReadCSV_EmployeeV1 ở trên với các gợi ý sau:

- Tổ chức mã nguồn bằng cách sử dụng các hàm riêng, các nghiệp vụ được tổ chức trong lớp thư viện riêng để sau này đóng gói và cung cấp thư viện back-end cho

nhóm làm giao diện. Cần các hàm với tham số truyền vào là danh sách nhân viên với cấu trúc đã được định nghĩa thích hợp cho bài toán. Trong đó sử dụng con trỏ (pointer) hoặc không sử dụng sao cho hợp lý (tối ưu bộ nhớ, tốc độ truyền dữ liệu) với các chức năng sau:

- Đọc file csv vào danh sách (bộ nhớ máy tính).
 - Thêm 1 nhân viên mới vào danh sách.
 - Xóa 1 nhân viên khỏi danh sách theo Mã nhân viên.
 - Cập nhật thông tin nhân viên trong danh sách.
 - Tìm kiếm nhân viên có mã số chứa một chuỗi kí tự cho trước, không phân biệt chữ hoa và chữ thường.
 - Lưu file csv ra thư mục được chỉ định.
 - Lấy danh sách nhân viên có ngày sinh nhật trong tuần tới (tuần được tính từ CN đến Thứ Hai).
- Chương trình hỗ trợ file CSV (dùng mã UTF-8 hợp lý).

Gợi ý:

- Sử dụng cấu trúc (struct) để mô tả dữ liệu của nhân viên.
- Sử dụng slice để chứa danh sách nhân viên.

Chương trình ReadCSV_EmployeeV2

Nâng cấp chương trình ReadCSV_EmployeeV1 ở trên với các gợi ý sau:

- Tổ chức mã nguồn bằng cách sử dụng các hàm riêng, các nghiệp vụ được tổ chức trong lớp thư viện riêng để sau này đóng gói và cung cấp thư viện back-end cho nhóm làm giao diện. Cần các hàm với tham số truyền vào là danh sách nhân viên với cấu trúc đã được định nghĩa thích hợp cho bài toán. Trong đó sử dụng con trỏ (pointer) hoặc không sử dụng sao cho hợp lý (tối ưu bộ nhớ, tốc độ truyền dữ liệu) với các chức năng sau:

Thử thách cho chương 3

Nếu bạn hoàn toàn chủ động làm được các chương trình sau và lưu kết quả lên `GITLAB.COM` thì có thể bỏ qua nội dung bài học trong ngày 4 này.

Chương trình ImportCSV2DB

Hãy viết chương trình để nạp dữ liệu từ 1 file CSV vào một cơ sở dữ liệu (CSLD) phổ biến như SQL Server, MySQL, PostgreSQL, SQLite.

Bạn tự tổ chức file CSV có đủ các tình huống sau:

- Kiểu dữ liệu các cột đa dạng như ví dụ sau (bạn có thể cho ví dụ khác):
 - ID: số nguyên định danh duy nhất của dòng dữ liệu.
 - Họ tên: kiểu chuỗi
 - Ngày tháng năm sinh: kiểu thời gian (ngày)
 - Trình độ học vấn: số nguyên (bạn tự quy định ý nghĩa)
 - Cấp bậc công việc: số thực (ví dụ trong công ty có dạng cấp bậc x.y như: 2.2; 3.3;....)
 - Giới tính: boolean (true: nam; false; nữ; rỗng là không xác định)
 - Tự giới thiệu: kiểu text với nội dung nhiều dòng; tối đa 1024 kí tự.
 - ...

Bạn tự tổ chức dự án trên GITLAB của bạn, commit và push liên tục sau mỗi lần bạn code được cái gì đó chạy được. Gợi ý có các tag để chốt các phiên bản như sau:

- Tag “v0.0.1” có chức năng tối thiểu:
 - Import được file 1 CSV đơn giản vào database cho trước.
 - Thông tin kết nối database được truyền từ dòng lệnh (gồm host, database type, port, database name, username, password,...)
- Tag “v0.0.2” có thêm các chức năng tối thiểu:
 - Import được 2 file CSV vào 2 table tương ứng trong database cho trước.
 - Thông tin kết nối database được bổ sung thêm: hỗ trợ truyền file cấu hình dạng .yaml

- Tag “v0.0.3” có thêm các chức năng tối thiểu:
 - Import được nhiều hơn 2 file CSV vào các table tương ứng trong database. Thông tin ánh xạ file CSV nào được nạp dữ liệu cho table nào được quy định trong một file cấu hình cũng ở dạng CSV (gồm 2 cột: CSV_File, Table_Name); tên file, hoặc đường dẫn đầy đủ của file csv mapping này được truyền từ tham số.

Chương trình DataTransform

Tình huống đặt ra là trong chương trình ImportCSV2DB ở trên, file dữ liệu CSV có cột “Ngày bắt đầu học GOLANG” được người dùng nhập tự do. Vì lý do nào đó mà hệ thống nhập liệu không qui định định dạng nên dữ liệu lưu trong file CSV có các dạng sau:

Dữ liệu	Ý nghĩa
31/12/2021	ngày/tháng/năm: DD/MM/YYYY
2021-12-31	năm-tháng-ngày: YYYY-MM-DD
20211231	nămthángngày: YYYYMMDD

Cột “Giới tính” cũng bị tình trạng như vậy, dữ liệu trong file CSV cũng có nhiều dạng như sau:

Dữ liệu	Ý nghĩa
true	nam
false	nữ
1	nam
0	
nam	nam
nữ	nữ
nu	nữ

(Dữ liệu giới tính có thể có cả chữ hoa xen lẫn chữ thường như: True, TRUE, NAM, naM, NU,...)

Nhiệm vụ của bạn:

- Triển khai bản binary của chương trình ImportCSV2DB để sẵn sàng sử dụng. Tùy theo tiến độ hoàn thành của các bạn thì Mentor có thể yêu cầu bạn hãy đóng gói sản phẩm từ mã nguồn của thành viên khác để sử dụng.

- Hãy chuẩn bị một CSDL để sử dụng phần mềm `ImportCSV2DB` để nạp dữ liệu file CSV ở trên vào với kiểu dữ liệu cho 2 cột “Ngày bắt đầu học GOLANG” và “Giới tính” là chuỗi (varchar trong CSDL)
- Sau khi đã có CSDL (gọi là CSDL nguồn) với data được nạp từ file CSV, hãy viết chương trình `DataTransform` có chức năng sau:
 - Đọc 2 cột dữ liệu không đúng kiểu ở trên từ CSDL nguồn để chuyển vào một CSDL đích với kiểu dữ liệu “Ngày bắt đầu học GOLANG” là kiểu ngày (date); “Giới tính” là kiểu boolean (hoặc kiểu số nguyên cũng được nếu bạn thích).
 - Hỗ trợ CSLD nguồn và CSDL đích là hai loại khác nhau.
 - Thông tin kết nối CSLD do bạn tự quy định trên tham số dòng lệnh, hoặc file cấu hình (khuyến khích dùng file .yaml); miễn là không được hard-code (code cứng trong mã nguồn).

Chương trình DataCRUDE

Viết chương trình dạng dòng lệnh cho phép người dùng chạy lệnh để thực hiện các như cầu sau với CSLD đích trong chương trình `DataTransform`:

- Cho file CSV mới với các dòng dữ liệu có ID đã tồn tại trong CSDL đích. Viết chức năng cho người dùng nạp dữ liệu từ file CSV này để cập nhật lại dữ liệu đang có trong CSDL. Trường hợp một dữ liệu trong file CSV mà chỉ có ID thôi thì có nghĩa là xóa dữ liệu với ID tương ứng trong CSDL đích.
- Cho phép người dùng chạy lệnh để export toàn bộ dữ liệu đang có trong CSDL ra file CSV (đường dẫn file được chỉ định trên tham số dòng lệnh).
- Xong các chức năng trên hãy tag mã nguồn với tên “v0.0.1”. Sau đó bổ sung thêm chức năng sau (tag với tên “v0.0.2” sau khi hoàn thành):
 - Cho phép người dùng khai báo câu truy vấn để export dữ liệu ra file CSV. Câu truy vấn có thể truyền trên tham số dòng lệnh, hoặc để sẵn trong file cấu hình.

Thử thách trong chương 4

Chương trình SumConsole

Hãy viết hàm Cộng hai số dạng chuỗi như yêu cầu ở đây https://thachln.github.io/chia-se/new-member/challenges/Add2Num_High-level-requirement_v1.6.pdf để đóng gói một thư viện (library/component) dùng chung.

Viết tiếp 2 chương trình dạng dòng lệnh (console) có tên SumCmd và 1 Web package dạng Restful API (tên package SumAPI, URL của API: sum) để nhận 2 chuỗi số từ bên ngoài. Chương trình SumCmd và SumAPI sử dụng file cấu hình chứa tên của function thực hiện chức năng cộng 2 số dạng chuỗi. Function này có trong thư viện binary ở trên. Tùy vào nội dung file cấu hình thì function có tên tương ứng trong thư viện binary được gọi để thực hiện thuật toán cộng.

Hãy áp dụng Unit Testing để kiểm tra thư viện của đồng đội và tóm tắt kết quả kiểm tra.

Gợi ý: Bạn cần tìm hiểu cách thức gọi hàm động nếu biết trước tên hàm (và các thông tin liên quan như package name, v.v...)