Kỹ Thuật Lập Trình

Lab 5 – Sử dụng mảng, chuỗi và cấu trúc

1 MỤC TIÊU CỦA BÀI THỰC HÀNH

- Định nghĩa được kiểu dữ liệu mới sử dụng typedef và struct.
- Khai báo được các biến có kiểu struct và array
- Ghi và đọc dữ liệu cho các thành phần của struct
- Ghi và đọc dữ liệu cho mảng
- Sử dụng struct và mảng để giải quyết một phần hoặc toàn bộ bài toán thực tế.
- Thực hiện các bài toán trên chuỗi

2 BÀI TẬP BẮT BUỘC

Câu 1:

- Định nghĩa **các kiểu dữ liệu mới** để có thể lưu trữ được thông tin của các thực thể (đối tượng) như sau. Với mỗi loại đối tượng như vậy, hãy tự xác định ít nhất 4 thông tin của nó, tự xác định kiểu dữ liệu của các trường thông tin này.
    - Máy in
    - Môn học
    - Doanh nghiệp
    - Con chim
- Khai báo một mảng cho mỗi loại thực thể ở trên. Mỗi mảng có thể chứa 100 thực thể. Hãy sử dụng macro cho con số 100.
- Thực hiện việc gán giá trị (xác định thông tin) của 1 thực thể bất kỳ trong 100 thực thể đó (làm cho từng loại nói trên)
- In ra thông tin thực thể vừa gán, cũng làm cho từng loại
- Khai báo một biến có kiểu là “Máy in” nói trên, khởi động với các giá trị bất kỳ để mô tả một máy tin cụ thể. Gán thực thể máy tin vừa tạo ra vào một trong 100 thực thể máy in ở trên, vị trí bất kỳ. Làm lại các công việc trong mục này cho các kiểu dữ liệu khác như: “Môn học”, “Doanh nghiệp”, v.v.

Hướng dẫn:

Sinh viên có thể tham khảo ví dụ sau về khai báo kiểu dữ liệu mới Máy tính với 2 thông tin: tên (chuỗi) và cân nặng (số thực):

```
typedef struct {
    string name;
    float resolution[2];
} computer;
```

Kiểu dữ liệu mới của chúng ta được định nghĩa dựa trên struct, nên cách khởi tạo sẽ giống struct:

Cách 1 (hợp lệ trong C):

```
computer com = {
    .name = "ASUS",
    .resolution = { 1366, 768 },
};
```

Hoặc nếu ta chỉ muốn khởi tạo giá trị cho trường thông tin "resolution" (không thể khởi tạo theo cách này trong C++):

```
computer com = {
    .resolution = { 1366, 768 },
};
```

Cách 2 (hợp lệ trong C++):

```
computer com = {
    "ASUS",
    { 1366, 768 },
};
```

Sinh viên có thể thử bằng cách dùng trình biên dịch online như CodeChef để thử các trường hợp:

- C (Gcc-4.9.2)
- C++ 4.9.2 (Gcc-4.9.2)
- C++ 14 (G++-4.9.2)

Cách gán cấu trúc tương tự như gán giá trị vào biến thông thường:

- com2 = com;
- com_array[10] = com;

Thành phần trong cấu trúc (struct) có thể được gán bằng câu lệnh có dạng như sau:

```
com.name = "My new computer";
```

Lưu ý: giá trị của vế phải trong phép gán phải tương thích với kiểu dữ liệu của trường thông tin trong cấu trúc.

Truy xuất trường thông tin trong đối tượng bằng cách:

```
var_struct.attr
```

với:

- var_struct: tên biến có kiểu dữ liệu là struct
- attr: trường thông tin cần truy xuất

Khai báo mảng có kích thước MAX (với MAX là hằng số):

```
computer com_array[MAX];
```

Ta có thể lấy thông tin của 1 đối tượng struct đang nằm trong 1 mảng bằng cách:

```
var_array [index].attr
```

với:

- var_array: tên biến kiểu mảng
- index: vị trí phần tử trong mảng

attr: trường thông tin cần truy xuất

Câu 2:

Viết chương trình cho phép:

- Đọc hai vector trong không gian N chiều từ tập tin đầu "input.txt". Gọi đó là vector **a** và **b**. N là số chiều của hai vector. Sinh viên phải xác định N khi đọc dữ liệu từ hàng đầu tiên. N không thể lớn hơn 50. Tập tin "input.txt" có định dạng như sau cho các vector trong không gian 3 chiều. Nếu quá trình đọc có lỗi hay hai vector không dùng số chiều thì in ra thông báo và dừng chương trình.

```
----------------------
----- Vectors ------
----------------------
12.4  34.2 44.2
3.5  2.21  21.56
```

- Tính và in ra:
  - ✓ Chiều dài của mỗi vector. Ví dụ: vetor a = [a1 a2 a3 a4] trong không gian 4 chiều có chiều dài là: sqrt(a1*a1 + a2*a2 + a3*a3 + a4*a4)
  - ✓ Tính và in ra **tích vô hướng** và **tích hữu hướng** (*chỉ tính tích hữu hướng cho trường hợp 3 chiều*) giữa hai vector.
    - **Cho biết:** Nếu có 2 vector **a** và **b**: $\mathbf{a}=[a_1, a_2, a_3]^T$ và $\mathbf{b}=[b_1, b_2, b_3]^T$, tích vô hướng và hữu hướng của **a** và **b** được tính như sau.
    - $\mathbf{a} \bullet \mathbf{b} = a_1b_1 + a_2b_2 + a_3b_3$
    - $\mathbf{a} \times \mathbf{b} = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix}\mathbf{i} - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix}\mathbf{j} + \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix}\mathbf{k}$
  - ✓ Tính và in ra góc tạo bởi hai vector.
  - ✓ Tính và in ra độ dài của hình chiếu vector **a** trên vector **b** và ngược lại
    - Hai câu cuối sử dụng công thức **a**.**b** = |**a**| * |**b**| * cos(**alpha**). **alpha** là góc tạo bởi hai vector a và b. **alpha** = acos(**a**.**b**)/(|**a**| |**b**|). Độ dài hình chiếu **a** trên **b** là: **a**.**b**/|**b**|

Yêu cầu cụ thể:

- Sinh viên phải sử dụng macro cho hằng số 50 nói trên
- Sinh viên phải định nghĩa một cấu trúc vector. Hãy tự xác định cần phải lưu trữ gì cho một vector.

Hướng dẫn:

a) Dữ liệu:

- Có thể định nghĩa một struc VECTOR có các phần tử như sau: value, dimension. Sinh viên xác định kiểu các phần tử cho phù hợp và thực hiện việc đặt tên biến sao cho gợi nhớ.
- Define các hằng số MAX_DIMENSION phục vụ cho việc nhập các phần tử từ input và PI, DEG dùng cho việc chuyển đổi góc từ radian sang độ.
- Khai báo các biến sử dụng cho đọc file input:
  - string str;
  - istringstream iss;
- Khai báo các biến liên quan đến tính toán vector:
  - double sum_square, scalar_p, angle …

b) Giải thuật:

- Giải thuật đọc file input:
  - Sử dụng hàm getline (cin, str) để đọc một line từ luồng nhập (file) và gán vào biến str.
  - Thực hiện việc getline để bỏ qua 3 dòng đầu của file input
  - Đọc vector 1.
  - Sử dụng hàm istringstream::str() để gán lại contain của iss. Việc này cho phép sử dụng lại được biến iss cho phép đọc các dòng khác trong file.
  - Sử dụng vòng lặp while để lần lượt dán giá trị cho vector. Điều kiện dừng while khi việc gán bị báo lỗi. (tương tự lab3)
  - Clear cờ lỗi của istringstream
  - Thực hiện tiếp cho vector 2

Câu 3:

Viết chương trình cho phép:

a) Nhập vào một dãy số thực không âm và lưu các số này vào mảng 1 chiều. Việc nhập chỉ kết thúc khi người dùng nhập bất kỳ số âm nào hoặc số lượng phần tử đọc vào đã là MAX_SIZE. Với MAX_SIZE là số nguyên dương được định nghĩa bởi chỉ thị #define

b) Tính và in ra: trung bình cộng và độ lệch chuẩn của dãy số đã nhập ở trên. Giả sử, mảng đã nhập có N phần tử, độ lệch chuẩn được tính theo công thức sau:

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

c) Đảo (không dùng mảng tạm) và in ra dãy số nói trên:

Hướng dẫn:

Chương trình nên in hướng dẫn cách nhập trước khi người dùng bắt đầu nhập.

Một trong những cách để giải câu a đó là:

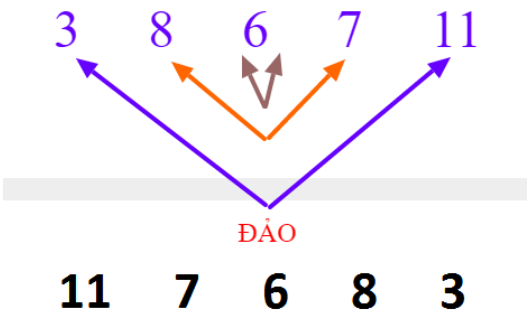
- Dùng vòng lặp for để lặp MAX_SIZE lần, ta sẽ dừng vòng lặp for sớm khi người dùng nhập vào số âm.
- Ta có thể lồng ghép thêm 1 vòng lặp while ở trong for để bắt các trường hợp nhập vào không hợp lệ (nhập chữ thay vì số), đây là tùy chọn.

Trong câu b, sinh viên cần viết vòng lặp để:

- Tính tổng các số trong mảng, sau đó tính trung bình bằng cách chia cho số phần tử hợp lệ có trong mảng (N <= MAX_SIZE).
- Tính tổng các bình phương như trong công thức độ lệch chuẩn đề cập, sau đó ta tiến hành tính độ lệch chuẩn.

Để đảo mảng mà không dùng mảng tạm, ta có thể thực hiện cách sau đây:

- Dùng vòng lặp *for* lặp N/2 lần (1 nửa số phần tử có trong mảng)
- Mỗi lần lặp ta đảo phần tử đang xét với phần tử tương ứng từ cuối đếm lên.



Ý tưởng tương tự có thể được hiện thực bằng vòng lặp *while*:

- Khởi tạo chỉ số **left** và **right** lần lượt là chỉ số của phần tử đầu tiên (tức là 0) và chỉ số của phần tử cuối cùng (tức là N-1) trong mảng.
- Mỗi lần lặp trong *while* ta đảo phần tử trong mảng có chỉ số **left** với phần tử có chỉ số **right**, sau đó **left** tiến 1 và **right** lùi 1.

Điều kiện dừng của vòng lặp là khi **left** và **right** vượt qua nhau, tức là **left > right**.

Câu 4:

a) Khai báo một cấu trúc có thể lưu giữ một điểm trong không gian 2 chiều.
b) Khai báo một mảng có thể lưu giữ danh sách toạ độ các đỉnh của một đa giác có tối đa MAX_VERT đỉnh. MAX_VERT là hằng số định nghĩa bởi #define
c) Viết chương trình cho phép người dùng nhập vào N là số đỉnh. 2 < N <= MAX_VERT. Nếu nhập không đúng thì yêu cầu nhập lại cho đến khi nhập đúng
d) Sinh ra ngẫu nhiên toạ độ của N đỉnh nói trên. Các toạ độ phải nằm trong khoảng -100 đến 100.
e) Giả sử, các toạ độ sinh ngẫu nhiên ở trên đã hình thành một đa giác. Tính và in ra chu vi của đa giác.

f) Chuyển hướng việc xuất ra tập tin "output.txt"

Hướng dẫn:

a) Dữ liệu: Chương trình phải có các biến (ô nhớ) để chứa các kiểu dữ liệu:

- **N**: kiểu float (để kiểm tra yêu cầu nhập vào chính xác n phải là một số nguyên không âm)

Khai báo kiểu dữ liệu **struct Vertice** bao gồm 2 biến để định nghĩa tọa độ của một đỉnh trong không gian 2 chiều (Phân biệt cách khai báo struct có sử dụng typedef và khi không sử dụng typedef)

- **x**: kiểu float
- **y**: kiểu float

Khai báo mảng một chiều **vertices[MAX_VERT]** với các phần tử có kiểu dữ liệu là một **struct Vertice** để lưu giữ danh sách tọa độ các đỉnh của một đa giác

b) Giải thuật:

- Kiểm tra điều kiện nhập vào của giá trị **N** nhập vào. Yêu cầu phải kiểm tra được **N** là số nguyên trong khoảng từ **3** đến **MAX_VERT**. Mọi giá trị nhập không thỏa điều kiện đều phải được nhập lại

- Tìm cách khởi tạo một số ngẫu nhiên cho **seed** ở mỗi lần thực thi khác nhau bằng hàm **srand()** (Tìm hiểu thêm số ngẫu nhiên giả (pseudo-random) để hiểu tại sao cần hiện thực như vậy)

- Sử dụng hàm **rand()** để sinh ra ngẫu nhiên tọa độ của các đỉnh trong khoảng -100 đến 100

- Do giả sử các tọa độ đã sinh ngẫu nhiên ở trên có thể hình thành một đa giác và các đỉnh liền kề tạo thành 1 cạnh của đa giác, nên sử dụng macro để định nghĩa phép toán tính khoảng cách (độ dài) của một cạnh tạo ra bởi 2 đỉnh liền kề, từ đó suy ra chu vi. Lưu ý khi sử dụng macro: tất cả các tham số của macro đều phải nằm trong cặp dấu (), khi sử dụng trong vế thay thế để tránh lỗi ngữ nghĩa về độ ưu tiên.

- Để chuyển hướng việc xuất ra tập tin "output.txt" thông qua command line, giả sử chương trình có tên Vertex.exe, để xuất ra tập tin "output.txt" sử dụng lệnh

Vertex.exe > output.txt

Câu 5:

Cho tập tin đầu vào chứa dữ liệu dạng ma trận, có định dạng như mô tả sau:

- Có N hàng dữ liệu. N từ 3 đến 100.
- Mỗi hàng có chứa M con số. M từ 2 đến 100

a. Viết chương trình nhập vector X và ma trận W từ tập tin đầu vào. Ở đó, X là cột dữ liệu đầu tiên của ma trận trong tập tin đầu vào. W là ma trận từ chứa M hàng mà (M-1) cột dữ liệu còn lại.

b. Tính tích của X*W và in ra kết quả.

Lưu ý: sử dụng macro cho các hằng số nói trên. Tự nhập tập in và kiểm tra kết quả.

Hướng dẫn:

a) Dữ liệu: Chúng ta cần dùng cấu trúc phù hợp để lưu các thông tin về Vector X, ma trận W. Chú ý về kích thước dữ liệu cần tạo sao cho có thể chứa được ma trận cũng như vector trong trường hợp M, N ở mức cao nhất.

b) Giải thuật:

- Để đọc file ta sẽ sử dụng kỹ thuật chuyển hướng nhập xuất để định hướng nội dùng file nào std::stdin và đọc bình thường như khi nhập từ màn hình. Ta nên đọc dòng đầu tiên trước để xác định được số cột của mỗi dòng. Đối với các hàng tiếp theo dùng hàm getline kết hợp vòng lặp để đọc từng dòng của file input. Trong mỗi vòng lặp ta sử dụng stringstream và một vòng lặp để bắt đầu đọc từng số trong một dòng và gán vào cấu trúc đã tạo trước đó, vòng lặp sẽ kết thúc khi dòng đã hết (dùng stringstream::eof để kiểm tra). Trong lúc đọc, kiểm tra số cột dữ liệu của mỗi dòng có khớp với hàng đầu tiên không, nếu không thì báo lỗi và thoát chương trình.
- Sau khi dữ liệu đã được đọc lên và lưu lại, ta sẽ biết được chính xác kích thước của vector và ma trận. từ đây ta sẽ biết được kích thước của kết quả phép nhân vector-ma trận X*W. Tích của vector và ma trận sẽ cho ra một vector K, trong đó giá trị của một phần tử ở vị trí i (K[i]) sẽ được tính bằng công thức sau
- $K[i] = \sum_{j=1}^{N} X[j] * W[i][j]$. Để tính giá trị này ta có thể dùng vòng lặp để cộng dồn các giá trị với nhau. Sinh viên có thể nhìn ví dụ sau cho dễ hình dung.

$$(1 \quad 2 \quad 3) \times \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix} = \begin{pmatrix} 1*1+2*2+3*3 \\ 1*4+2*5+3*6 \\ 1*7+2*8+3*9 \end{pmatrix}^T = \begin{pmatrix} 14 \\ 32 \\ 50 \end{pmatrix}^T$$

- Sinh viên lựa định dạng phù hợp để in ra các giá trị trong vector kết quả K. Nhớ kết hợp với việc canh lề và thiết lập độ chính xác để kết quả được đẹp.

Câu 6:

Viết chương trình nhập vào một chuỗi. Xác định chiều dài của chuỗi. Nghĩa là, hiện thực lại hàm strlen trong <string.h>.

Hướng dẫn:

a) **Dữ liệu:** Chương trình phải có các biến (ô nhớ) để chứa các kiểu dữ liệu:
   - Biến chứa chuỗi nhập vào từ bàn phím: khai báo mảng kí tự
   - Biến chứa độ dài chuỗi: kiểu dữ liệu size_t

b) **Giải thuật:**
   - Dùng macro tiền xử lý #define để chỉ rõ kích thước bộ đệm lưu chuỗi kí tự nhập vào từ bàn phím
   - Dùng hàm cin.getline để lấy chuỗi kí tự nhập từ bàn phím (không dùng hàm cin vì cin không lấy được kí tự khoảng trắng từ bàn phím)
   - Duyệt và đếm từng kí tự của chuỗi nhập vào cho đến khi bắt gặp kí tự '\0' (kí tự kết thúc chuỗi, độ dài chuỗi không tính kí tự này)
   - Số kí tự đếm được chính là độ dài của một chuỗi

Câu 7:

Viết chương trình nhập vào một chuỗi. Xoá khoảng trắng hai đầu và chỉ giữ lại 01 khoảng trắng giữa các từ.

Hướng dẫn:

a) **Dữ liệu:** Chương trình phải có các biến (ô nhớ) để chứa các kiểu dữ liệu:
   - Biến chứa chuỗi nhập vào từ bàn phím: khai báo mảng kí tự

b) **Giải thuật:**
   - Dùng macro tiền xử lý #define để chỉ rõ kích thước bộ đệm lưu chuỗi kí tự nhập vào từ bàn phím
   - Dùng hàm cin.getline để lấy chuỗi kí tự nhập từ bàn phím (không dùng hàm cin vì cin không lấy được kí tự khoảng trắng từ bàn phím)
   - Duyệt qua các khoảng trắng ở giữa hai từ liên tiếp trong chuỗi và đếm số lượng khoảng trắng
   - Xóa các khoảng trắng ở giữa hai từ liên tiếp vừa tìm được (giữ một khoảng trắng ở giữa hai từ liên tiếp).

- Lặp lại bước 3 và 4 cho đến khi kết thúc chuỗi.
  Ví dụ: “ Lap trinh C++ “ -> “Lap trinh C++ “ ->
  “Lap trinh C++ “ -> “Lap trinh C++ “ -> “Lap trinh C++“

Câu 8:

Viết chương trình nhập vào một chuỗi. Đảo các từ trong chuỗi. In ra chuỗi ban đầu và chuỗi đã thay đổi.

Hướng dẫn:

a) **Dữ liệu:** Chương trình phải có các biến (ô nhớ) để chứa các kiểu dữ liệu:
   - Biến chứa chuỗi nhập vào từ bàn phím: khai báo mảng kí tự

b) **Giải thuật:**
   - Dùng macro tiền xử lý #define để chỉ rõ kích thước bộ đệm lưu chuỗi kí tự nhập vào từ bàn phím.
   - Dùng hàm cin.getline để lấy chuỗi kí tự nhập từ bàn phím (không dùng hàm cin vì cin không lấy được kí tự khoảng trắng từ bàn phím).
   - Đảo ngược toàn bộ kí tự trong chuỗi
     Ví dụ: “diep thanh dang” -> “gnad hnaht peid”
   - Đảo ngược các kí tự trong từng từ trong chuỗi
     Ví dụ: “gnad hnaht peid” -> “dang thanh diep”