

## BÀI THỰC HÀNH SỐ 12 :: BÀI TẬP TỔNG HỢP

### MỤC TIÊU

1. Vận dụng các cấu trúc điều khiển rẽ nhánh và lặp để diễn đạt các ngữ cảnh của bài toán cần giải quyết bằng ngôn ngữ lập trình C.
2. Phân rã bài toán cần giải quyết thành các bài toán nhỏ hơn và xây dựng các chương trình con để giải quyết từng vấn đề nhỏ.
3. Sử dụng cấu trúc mảng một chiều, tổ chức lưu trữ dữ liệu cho các bài toán có dữ liệu tuyến tính và thực hiện giải quyết các công việc của một bài toán tin học với 2 thao tác cơ bản là Duyệt, Tìm kiếm và Sắp xếp.

### MỘT SỐ BÀI TOÁN ĐỀ XUẤT

#### Vấn đề 1 (*Cấu trúc rẽ nhánh*): Đọc số thành chữ

Một công việc anh/chị thường làm khi thực hiện các giao dịch về tiền tệ là bên cạnh việc viết số tiền bằng số (chẳng hạn: 18.500 VND) thì còn phải viết số tiền bằng chữ (mười tám ngàn năm trăm đồng). Hai giao dịch dưới đây, một là giao dịch chuyển khoản sử dụng ebanking của VCB có ghi rõ số tiền bằng số và số tiền bằng chữ và giao dịch hóa đơn của quán cà phê còn thiếu chức năng đọc số tiền bằng chữ. Nhiệm vụ của các anh/chị ở bài toán này là viết chương trình đọc số thành chữ, với yêu cầu số có thể có đến 9 chữ số.

GIAO DỊCH

Nhà cung cấp dịch vụ

FPT ADSL (Miền Bắc)

Mã khách hàng

HNFD94864

Tên khách hàng

Nguyen Dinh Hung

Thông tin thêm

Mô tả

MKH : HNFD94864.Bill Cycle : ;31/08/2017;..

Số tiền thanh toán

181,500 VND

Số tiền bằng chữ

Một trăm tám mươi một ngàn năm trăm đồng

Số lệnh giao dịch

1009170954395001

Thời gian

10/09/2017 07:35:29

Giao dịch thành công!

Cảm ơn Quý khách đã sử dụng dịch vụ của Vietcombank

Cafe Acoustic

116 Lý Thường Kiệt - TP Buôn Ma Thuột

PHIẾU THANH TOÁN

BÀN SỐ 11

Stt	Tên	Đơn giá	SL	T. tiền
1	Cafe đen nóng	6,000	3	18,000
2	Cafe kem	17,000	1	17,000
3	Bí đao	12,000	1	12,000
4	sữa nước sôi	8,000	1	8,000
5	Nước me	12,000	1	12,000
Tổng cộng:				67,000
Giảm giá:				0
Còn lại:				67,000

BMT, 22 tháng 10 năm 2011

Người thu tiền

Thu Ngân

Xin hẹn gặp lại!

Yêu cầu: Nhận vào một số nguyên không âm. In ra màn hình cách đọc số đó trong tiếng Việt hoặc tiếng Anh. Chẳng hạn: 104 đọc là “một trăm linh bốn” hoặc “một trăm lẻ bốn” nhưng không đọc là “một trăm không bốn”. Và 114 thì đọc là “một trăm mười bốn” chứ không đọc là “một trăm mười tư”.

**Vấn đề 2 (Cấu trúc lặp và Hàm): Tìm giá trị của số Pi với độ chính xác  $\varepsilon$** 

Số **pi** (ký hiệu:  $\pi$ ) là một hằng số toán học có giá trị bằng tỷ số giữa chu vi của một đường tròn với đường kính của đường tròn đó. Hằng số này có giá trị xấp xỉ bằng 3,1415926535897. Nó được biểu diễn bằng chữ cái Hy Lạp  $\pi$  từ giữa thế kỷ XVIII.

$\pi$  là một số vô tỉ, nghĩa là nó không thể được biểu diễn chính xác dưới dạng tỉ số của hai số nguyên. Nói cách khác, nó là một số thập phân vô hạn không tuần hoàn. Hơn nữa,  $\pi$  còn là một số siêu việt - tức là nó không phải là nghiệm của bất kì đa thức với hệ số hữu tỉ nào. Các con số trong biểu diễn thập phân của  $\pi$  dường như xuất hiện theo một thứ tự ngẫu nhiên, mặc dù người ta chưa tìm được bằng chứng nào cho tính ngẫu nhiên này.

Trong hàng ngàn năm, các nhà toán học đã nỗ lực mở rộng hiểu biết của con người về số  $\pi$ , đôi khi bằng việc tính ra giá trị của nó với độ chính xác ngày càng cao. Trước thế kỷ XV, các nhà toán học như Archimedes và Lưu Huy đã sử dụng các kỹ thuật hình học, dựa trên đa giác, để ước lượng giá trị của  $\pi$ . Bắt đầu từ thế kỷ XV, những thuật toán mới dựa trên chuỗi vô hạn đã cách mạng hóa việc tính toán



số  $\pi$ , và được những nhà toán học như Madhava của Sangamagrama, Isaac Newton, Leonhard Euler, Carl Friedrich Gauss, và Srinivasa Ramanujan sử dụng.

Trong thế kỷ XXI, các nhà toán học và các nhà khoa học máy tính đã khám phá ra những cách tiếp cận mới - kết hợp với sức mạnh tính toán ngày càng cao - để mở rộng khả năng biểu diễn thập phân của số  $\pi$  tới  $10^{13}$  chữ số. Tháng 10 năm 2014, kỷ lục này được nâng lên 13.300.000.000.000 chữ số bởi một nhóm nghiên cứu lấy tên là *houkouonchi*. Các ứng dụng khoa học thông thường yêu cầu không quá 40 chữ số của  $\pi$ , do đó động lực của những tính toán này chủ yếu là tham vọng của con người muốn đạt tới những kỉ lục mới, nhưng những tính toán đó cũng được sử dụng để kiểm tra các siêu máy tính và các thuật toán tính nhân với độ chính xác cao.

**Yêu cầu:** *Tìm số  $\pi$  với độ chính xác càng cao càng tốt và thử thách cần hướng đến là tìm được 40 chữ số của  $\pi$  ở sau dấu chấm thập phân.*

**Hướng dẫn:** *Nhóm anh/chị hãy cài đặt tìm số  $\pi$  theo cả 3 phương pháp dưới đây. Và nhận xét phương pháp nào tốt nhất (theo nghĩa, tìm được nhiều chữ số của  $\pi$  với thời gian chấp nhận được)*

**Phương pháp 1:** *Tìm số  $\pi$  bằng cách sử dụng thuật toán dựa trên chuỗi vô hạn. Có rất nhiều chuỗi số có mối quan hệ với  $\pi$ , chẳng hạn:*

$$\frac{\pi}{4} \left[ \approx 1 - \frac{1}{3} + \frac{1}{5} - \dots + (-1)^n \frac{1}{2n+1} + \dots \right]$$

Hãy xây dựng hàm để tính số  $\pi$  theo công thức trên, quá trình lặp sẽ dừng khi  $\left|(-1)^n \frac{1}{2n+1}\right| < \varepsilon$

với  $\varepsilon$  là số đủ bé cho trước.

Phương pháp 2: Sử dụng Thuật toán lặp Gauss-Legendre.

Thuật toán này chấm dứt sự phụ thuộc vào các chuỗi vô hạn. Thuật toán lặp (iterative algorithm) lặp lại một phép tính đặc trưng, mỗi lần lặp lại sử dụng đầu ra từ bước lặp trước làm đầu vào của nó, và sinh ra một kết quả trong mỗi bước hội tụ về giá trị mong muốn. Phương pháp này gọi là phương pháp AGM (*arithmetic-geometric mean method*, phương pháp trung bình hình học-đại số) hay thuật toán Gauss-Legendre. Vì được sửa đổi bởi Salamin và Brent, nó cũng còn được gọi là thuật toán Brent-Salamin. Xem mã giả của thuật toán Gauss.

Thuật toán lặp Gauss-Legendre:

Khởi tạo

$$a_0=1 \quad b_0=\frac{1}{\sqrt{2}} \quad t_0=\frac{1}{4} \quad p_0=1$$

Lặp

$$a_{n+1}=\frac{a_n+b_n}{2} \quad b_{n+1}=\sqrt{a_n b_n}$$

$$t_{n+1}=t_n-p_n(a_n-a_{n+1})^2 \quad p_{n+1}=2p_n$$

Sau đó một phép ước lượng  $\pi$  được tính từ

$$\pi \approx \frac{(a_n+b_n)^2}{4t_n}$$

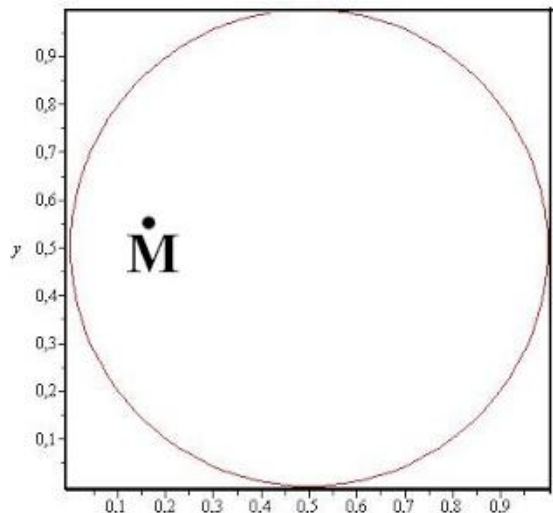
Phương pháp 3: Sử dụng phương pháp tính xác, cách tìm  $\pi$  theo phương pháp này như sau: Cho hình vuông cạnh 1 và một đường tròn nội tiếp hình vuông. Gieo một điểm ngẫu nhiên trong hình vuông. Lúc đó xác suất điểm này rơi vào trong hình tròn theo quan điểm hình học là  $\pi/4$ . Dựa vào kết quả này, ta thực hiện mô phỏng, gọi P là tần suất các điểm rơi vào hình tròn. Lúc đó số  $\pi$  được xấp xỉ bằng  $4P$ .

Giải thích: Gọi  $S_1$  là số lần viên sỏi rơi vào hình tròn và S là số lần gieo viên sỏi (chắc chắn viên sỏi nằm trong hình vuông). Khi đó:

$$\frac{S_1}{S} = \frac{\text{diện tích hình tròn}}{\text{Diện tích hình vuông}} = \frac{\pi x (\frac{1}{2})^2}{1 \times 1} = \frac{\pi}{4}$$

Hướng dẫn: Để việc tìm  $\pi$  càng chính xác thì chúng ta cần phải thực hiện càng nhiều lần gieo điểm càng tốt. Anh/chị được khuyến khích xây dựng các hàm dưới đây để thực hiện tìm pi theo phương pháp xác suất:

- *Gieodiem()*: Sinh ngẫu nhiên 1 điểm thuộc hình vuông.
- *Sukien()*: trả về 1 nếu điểm rơi vào hình tròn và trả về 0 nếu rơi ngoài hình tròn.
- *Solan()*: trả về tổng số lần gieo điểm
- *Tansuat()*: trả về tỷ số  $S_1/S$
- *Pi()* số  $\pi$  tìm được với sai số  $\varepsilon$



**Vấn đề 3 (Dữ liệu kiểu mảng): Kể chuyện về dữ liệu**

Điểm môn học **Nhập môn Lập trình** của sinh viên bao gồm 5 bài kiểm tra, để quản lý điểm của lớp, giáo viên dùng 5 mảng 1 chiều  $T_1, T_2, T_3, T_4, T_5$  lưu điểm của 5 bài kiểm tra tương ứng. Trong đó điểm là một con số nguyên thuộc đoạn  $[0, \dots, 10]$  và mỗi lớp có không quá 30 sinh viên. Anh/chị được yêu cầu thực hiện các công việc dưới đây:

1. Đọc dữ liệu từ bàn phím hoặc từ file văn bản, sau đó hiển thị thông tin điểm của từng sinh viên lên màn hình một cách trực quan bằng việc sử dụng các ký tự '\*' để vẽ biểu đồ điểm của từng sinh viên, sau khi vẽ xong biểu đồ điểm của mỗi sinh viên thì dừng lại đợi ấn enter mới đi tiếp. Chẳng hạn sinh viên A có điểm 5 bài kiểm tra lần lượt là: 5, 6, 8, 8, 10 thì biểu đồ điểm có dạng:

```
Test 1: * * * * *
Test 2: * * * * *
Test 3: * * * * *
Test 4: * * * * *
Test 5: * * * * *
```

2. Sắp xếp điểm của các bài kiểm tra theo thứ tự tăng dần, sau đó vẽ biểu đồ điểm của cả lớp theo từng bài kiểm tra, sau khi vẽ xong biểu đồ của 1 môn thì dừng lại đợi enter rồi đi tiếp. Chẳng hạn dưới đây là biểu đồ điểm *Test1* của lớp có 22 sinh viên với các điểm sau khi được sắp tăng: {2, 3, 3, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 8, 8, 9, 10}
3. Đếm số lượng sinh viên đạt các mức điểm từ 0 đến 10, sau đó vẽ biểu đồ theo số liệu thống kê này. Chẳng hạn với điểm *Test1* của lớp sinh viên trên chúng ta thống kê được dữ liệu ở bảng và biểu đồ tương ứng của nó như bên cạnh. Sau khi vẽ biểu diễn của mỗi cột điểm thì dừng lại để xem rồi ấn enter để đi tiếp (đến cột điểm tiếp theo).

```
1: * *
2: * * *
3: * * *
4: * * * *
5: * * * *
6: * * * *
7: * * * * *
8: * * * * *
9: * * * * *
10: * * * * *
11: * * * * *
12: * * * * *
13: * * * * *
14: * * * * *
15: * * * * *
16: * * * * *
17: * * * * *
18: * * * * *
19: * * * * *
20: * * * * *
21: * * * * *
22: * * * * *
```

Điểm	0	1	2	3	4	5	6	7	8	9	10
Số lượng	0	0	1	2	3	5	4	3	2	1	1

```
1:
2:
3: *
4: * *
5: * * *
6: * * * *
7: * * * *
8: * * * *
9: * *
10: *
```

4. Từ dữ liệu điểm của sinh viên trong lớp, thông qua thống kê mô tả sẽ giúp giảng viên phát hiện được nhiều điều thú vị về mối liên hệ giữa các cột điểm thi của một sinh viên và điểm thi của các sinh viên với nhau. Anh/chị hãy tìm hiểu các khái niệm *Trung bình (mean)*, *Trung vị (Median)*, *Yếu vị (Mode)*, *Khoảng biến thiên (Range)*, *Phương sai* và *Độ lệch chuẩn* xem ý nghĩa của mỗi thông số nói lên điều gì? Từ đó viết chương trình tính giá trị của chúng đối với điểm của từng sinh viên và đối với từng cột điểm của cả lớp.