

YÊU CẦU SINH VIÊN ĐỌC KỸ TRƯỚC KHI LÀM BÀI:

1. Sinh viên ghi họ tên, mã số sinh viên (MSSV) lên đề thi và phải nộp lại đề thi trước khi rời phòng thi. Không nộp lại đề sẽ nhận điểm 0.
2. Bài thi gồm có 2 câu. Khi nộp bài, sinh viên **CHỈ** nộp lại các file có tên *cau1.c*, *cau2.c* tương ứng với bài làm của Câu 1, Câu 2.
3. Khi nộp bài, sinh viên bắt buộc phải tạo một thư mục có tên là MSSV_HoTen (viết liền, không dấu) của mình, sau đó chép (copy) file *cau1.c*, *cau2.c* vào thư mục này và nộp bài (Không chép thêm các file nào khác). Ví dụ: SV tên là Nguyễn Văn An, có mã số sinh viên là 51702341 thì thư mục đặt tên là **51702341_NguyenVanAn**. Bên trong thư mục này có file **cau1.c** và **cau2.c**.
4. Bài thi sẽ được chấm bằng cách nhập và kiểm tra kết quả in ra màn hình. **Tất cả các trường hợp bị lỗi chương trình hoặc làm không đúng yêu cầu đều sẽ không được tính điểm.**

CÂU HỎI THI: SV thực hiện các câu hỏi sau trên ngôn ngữ lập trình C. Với mỗi câu hỏi, SV có thể viết dạng **function** và dùng hàm **main** để gọi các **function** nhằm thực hiện yêu cầu của đề bài.

Câu 1 (5.0 điểm). Viết chương trình nhập vào hai số nguyên dương m, h . Trong đó, h là số giờ làm việc của một nhân viên bán hàng trong một tuần và m là số tiền tiêu chuẩn được chi trả cho mỗi giờ làm việc. Giả sử người dùng luôn nhập m, h hợp lệ (không cần kiểm tra giá trị của m, h khi viết chương trình). Tính và in ra màn hình tổng số tiền mà nhân viên thu nhập được trong tuần. Cho biết cách tính như sau:

- Trong 40 giờ đầu tiên, mỗi giờ được trả theo số tiền tiêu chuẩn đã nhập.
- 5 giờ tiếp theo (nếu có vượt quá 40 giờ), mỗi giờ vượt (và nhỏ hơn hoặc bằng 45 giờ) được trả bằng 1.8 lần số tiền tiêu chuẩn.
- 5 giờ kế tiếp (nếu có vượt quá 45 giờ), mỗi giờ vượt (và nhỏ hơn hoặc bằng 50 giờ) được trả bằng 2.5 lần số tiền tiêu chuẩn.
- Nếu vượt quá 50 giờ, mỗi giờ vượt được trả bằng 2.5 lần số tiền tiêu chuẩn.

Ví dụ: $h = 35, m = 80 \Rightarrow$ Số tiền: $35 \cdot 80 = 2800$

$h = 47, m = 80 \Rightarrow$ Số tiền: $40 \cdot 80 + 5 \cdot 1.8 \cdot 80 + 2 \cdot 2.5 \cdot 80 = 4120$

Ghi chú: Dấu * là *phép toán nhân*.

Câu 2 (5.0 điểm). Chương trình tính phí dịch vụ taxi theo chiều dài đường đi d như sau:

- $d \leq 1$ km : giá 18 000 VND/km,
- $1 < d \leq 10$ km : giá 8 000 VND/km,
- $10 < d \leq 30$ km : giá 6 000 VND/km,
- $d > 30$ km : giá là 4 000 VND/km.

Ví dụ: nếu khách hàng đi 35km thì $Tiền = 1 \cdot 18000 + 9 \cdot 8000 + 20 \cdot 6000 + 5 \cdot 4000$

Câu 3 (5.0 điểm). Viết chương trình nhận vào tọa độ tâm $O(x_0, y_0)$ (số thực) và bán kính R (số thực) của một đường tròn, tọa độ điểm $A(x_A, y_A)$ (x_A, y_A là số thực). Chương trình kiểm tra điểm $A(x_A, y_A)$ nằm **trên**, **trong** hay **ngoài** đường tròn. Gọi ý: tính khoảng cách giữa điểm cần so sánh với tâm đường tròn. Sau đó so sánh khoảng cách này với bán kính đường tròn. Công thức tính khoảng cách giữa hai điểm A và O như sau:

$$d(A, O) = \sqrt{(x_A - x_O)^2 + (y_A - y_O)^2}$$

Câu 4 (5.0 điểm). Chương trình có các chức năng sau đây:

- Cho phép người dùng nhập 3 số biểu diễn độ dài 3 cạnh của một tam giác. Nếu độ dài không thỏa mãn điều kiện hình thành tam giác thì thông báo lỗi và kết thúc chương trình.
- Tính và in ra màn hình chu vi và diện tích của tam giác. Biết nửa chu vi và diện tích tam giác được tính như sau:

$$p = \frac{a + b + c}{2}$$

$$s = \sqrt{p(p-a)(p-b)(p-c)}$$

- Kiểm tra thuộc tính tam giác có thuộc tính cân, đều, hay thường.

Câu 5 (5.0 điểm). Dùng cấu trúc switch-case và if/else viết chương trình hỗ trợ máy bán hàng tự động.

Máy có thể bán 5 loại nước giải khát, mỗi loại có giá như mô tả sau:

Coca:	3000
Trà xanh:	6000
Sting:	4000
Sữa tươi:	5000
Nước suối:	3000

Người mua bỏ tiền xu vào máy, và nhấn các nút từ 1 tới 5 tương ứng với từng loại nước giải khát. Chương trình sẽ xuất ra tên loại nước giải khát đã mua. Máy có kiểm tra tiền bỏ vào đủ hay thiếu. Nếu dư thì thông báo tiền dư, thiếu thì thông báo lỗi và kết thúc chương trình.

Câu 6 (5.0 điểm). Viết chương trình nhập vào số nguyên $n > 0$ (nếu người dùng nhập vào không phải là số dương thì yêu cầu nhập lại), tính và in ra màn hình giá trị biểu thức sau (không dùng công thức tổng quát):

a.

$$S = \frac{2}{\sqrt{1}} + \frac{3}{\sqrt{2}} + \frac{4}{\sqrt{3}} + \frac{5}{\sqrt{4}} + \dots + \frac{n+1}{\sqrt{n}}$$

Ví dụ:

$n = 0 \Rightarrow$ Nhập lại n

$n = -10 \Rightarrow$ Nhập lại n

$n = 1 \Rightarrow S = 2$

$n = 2 \Rightarrow S = 4.1213$

$n = 10 \Rightarrow S = 27.4893$

b. $S_1 = 1 + 2 + \dots + n$

c. $S_2 = 1 + \frac{1}{2} + \dots + \frac{1}{n}$

d. $S_3 = 1 * 2 * \dots * n = n!$

e. $S_4 = x^y$

f. $S_5 = -1 + \frac{1}{2} - \frac{1}{3} + \frac{1}{4} - \dots + \frac{(-1)^n}{n}$

Với n, x, y là các số nguyên dương nhập từ bàn phím. Nếu người dùng nhập số nguyên âm thì yêu cầu nhập lại cho đến khi thỏa yêu cầu.

Câu 7 (5.0 điểm). Viết chương trình yêu cầu người dùng nhập một số nguyên dương. Chương trình kiểm tra nếu số không nguyên dương thì yêu cầu người dùng nhập lại (tối đa 5 lần). Khi số thỏa điều kiện thì:

- Tính và in ra số đảo của số đã nhập. Số đảo của 1234 là 4321.
- Kiểm tra số đó có là số hoàn thiện ([*perfect number*](#)) hay không. Một số là số hoàn thiện khi tổng các ước số (trừ số đó) của nó bằng chính nó. Số 6 là số hoàn thiện vì $1 + 2 + 3 = 6$. Số 28 là số hoàn thiện vì $1 + 2 + 4 + 7 + 14 = 28$.
- Kiểm tra số đó có là số đối xứng hay không. Các số 12321, 1221, 121 là số đối xứng. Số 12312, 124 không phải là số đối xứng. Gợi ý: số đảo của một số bằng chính nó \Rightarrow số đối xứng.

- d. Kiểm tra số đó có là số nguyên tố hay không. Một số là số nguyên tố khi nó chỉ chia hết cho 1 và chính nó. Số 2 là số nguyên tố bé nhất. Nếu số đó là số nguyên tố thì in số đó. Ngược lại, in tất cả số nguyên tố nhỏ hơn n .
- e. Kiểm tra số đó có phải là số Armstrong (Armstrong numbers) hay không. Số m là số Armstrong khi tổng lập phương các ký số của nó bằng chính nó. Ví dụ các số sau đây là số Armstrong:

$$153 = 1^3 + 5^3 + 3^3$$

$$370 = 3^3 + 7^3 + 0^3$$

Câu 8 (5.0 điểm). Viết chương trình yêu cầu người dùng nhập 2 số nguyên dương. Chương trình kiểm tra nếu một trong hai số không nguyên dương thì yêu cầu người dùng nhập lại. Khi hai số thỏa điều kiện thì:

- Tìm và in tất cả ước chung của hai số.
- Tìm và trả về ước chung lớn nhất của hai số.
- Tìm và trả về bội chung nhỏ nhất của hai số.
- Kiểm tra hai số này có phải là cặp số bạn ([*Amicable numbers*](#)) hay không. Hai số là cặp số bạn khi tổng các ước của số này bằng số kia và ngược lại. Cặp (220, 284) là cặp số bạn vì các ước số của 220 là 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110. Tổng các giá trị này đúng bằng 284. Các ước số của 284 là 1, 2, 4, 71, 142. Tổng các giá trị này đúng bằng 220.

Ghi chú: đây chỉ là một số dạng câu hỏi mẫu, đề thi thật sẽ có 2 câu có dạng tương tự như các câu hỏi này nhưng có chỉnh một ít.