NỘI DUNG ÔN TẬP GIỮA KỲ
MÔN PHƯƠNG PHÁP LẬP TRÌNH

Lưu ý:

- Sinh viên thực hiện các câu hỏi sau trên ngôn ngữ lập trình C.
- Bài làm được **biên dịch** bằng **câu lệnh gcc** trên **môi trường *nix** (Unix).
- Chỉ những bài **biên dịch thành công mới được chấm điểm**, bài **biên dịch không được sẽ không được chấm điểm** (0 điểm).

Câu 1: 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, mỗi giờ vượt được trả bằng 1.8 lần số tiền tiêu chuẩn.
- 5 giờ kế tiếp, mỗi giờ vượt được trả bằng 2.5 lần số tiền tiêu chuẩn.
- Nếu vượt quá 50 giờ, mỗi giờ vượt được trả bằng 2.6 lần số tiền tiêu chuẩn.

Ví dụ: *h* = 35, *m* = 80 => Số tiền: 35*80 = 2800

h = 47, *m* = 80 => Số tiền: 40*80 + 5*1.8*80 + 2*2.5*80 = 4320 (*dấu * là phép toán nhân*)

Câu 2: 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 35 km thì tiền = $1 * 18000 + 9 * 8000 + 20 * 6000 + 5 * 4000$

Câu 3: Viết chương trình nhập vào tọa độ tâm $O(x_0, y_0)$ (số thực) và bán kính R (số thực) của một đường tròn, tọa độ điểm $A(x_A, y_A)$ (số thực). Chương trình kiểm tra điểm $A(x_A, y_A)$ nằm **trên**, **trong** hay **ngoài** đường tròn. Gợi 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:

$$d(A,O) = \sqrt{(x_A - x_0)^2 + (y_A - y_0)^2}$$

Câu 4: Viết chương trình có các chức năng sau đây:

a. 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.

b. Tính và in ra màn hình chu vi và diện tích của tam giác.

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

Câu 5: Viết chương trình cho phép nhập vào giá trị x và y. Sau đó tính và in kết quả của biểu thức sau ra màn hình.

$$S = \left(3x^3y - \frac{1}{2}x^2 + \frac{1}{5}xy\right)6xy^3$$

Câu 6: Viết chương trình cho phép người dùng nhập vào một số thực x, hãy tính và in ra màn hình giá trị của hàm f(x) như sau:

$$f(x) = \begin{cases} \sin(x)\cos(5x) & nếu\ x < 0 \\ 5^x & nếu\ x = 0 \\ e^x & nếu\ 0 < x < 5 \\ \dfrac{5^x}{x+5} & nếu\ x \geq 5 \end{cases}$$

Câu 7: 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:

Tên sản phẩm	Đơn giá
Pepsi	6000
Trà xanh	7000
Sting	8000
Sữa tươi	6000
Nước suối	4000

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 8: 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. $S1 = \frac{2}{\sqrt{1}} + \frac{3}{\sqrt{2}} + \frac{4}{\sqrt{3}} + \cdots + \frac{n+1}{\sqrt{n}}$

Ví dụ:- `n = 0 -> Nhập lại n`

- `n = -10 -> Nhập lại n`
- `n = 1 -> S = 2`
- `n = 2 -> S = 4.1213`
- `n = 10 -> S = 27.4893`

b. $S2 = 1 + 2 + 3 + \cdots + n$

c. $S3 = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$

d. $S4 = 1 * 2 * 3 * \ldots * n = n!$

e. $S5 = -1 + \frac{1}{2} - \frac{1}{3} + \cdots + \frac{(-1)^n}{n}$

Câu 9: 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 (chỉ được nhập lại tối đa 5 lần). Khi giá trị nhập thỏa điều kiện thì thực hiện các yêu cầu sau:

a. Tính và in ra số đảo của số đã nhập. Ví dụ: Số đảo của 1234 là 4321.
b. 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.
c. Kiểm tra số đó có là số đối xứng hay không. VD: 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 => 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 (từ 1 đến n).
e. Kiểm tra số đó có phải là số Armstrong (Armstrong numbers) hay không. Số n 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 10: 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:

a. Tìm và in tất cả ước chung của hai số.
b. Tìm và trả về ước chung lớn nhất của hai số.
c. Tìm và trả về bội chung nhỏ nhất của hai số.
d. 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.