

BÀI TẬP

1. Có thể truy nhập thành phần của cấu trúc thông qua con trỏ như sau (với p là con trỏ cấu trúc và a là thành phần của cấu trúc):

A: (*p).a B: *p→a C: a và b sai D: a và b đúng

2. Cho khai báo struct T {int x; float y;} t, *p, a[10]; Câu lệnh nào trong các câu sau là không hợp lệ:

(1) p = &t; (2) p = &t.x; (3) p = a;
(4) p = &a (5) p = &a[5]; (6) p = &a[5].y;

A: 1, 2 và 3 B: 4, 5 và 6 C: 1, 3 và 5 D: 2, 4 và 6

3. Cho các khai báo sau:

```
struct ngay {int ng, th, nam;} vaotruong, ratruong;  
typedef struct {char hoten[25]; ngay ngaysinh;} sinhvien;
```

Hãy chọn câu đúng nhất

A: Không được phép gán: ratruong = vaotruong;

B: sinhvien là tên cấu trúc, vaotruong, ratruong là biến cấu trúc

C: Có thể viết: vaotruong.ng, ratruong.th, sinhvien.vaotruong.nam để truy nhập đến các thành phần tương ứng.

D: a, b, c đúng

4. Trong các khởi tạo giá trị cho các cấu trúc sau, khởi tạo nào đúng:

```
struct S1 {  
    int ngay, thang, nam;  
} s1 = {2,3};  
struct S2 {  
    char hoten[10];  
    struct S1 ngaysinh;  
} s2 = {"Ly Ly", 1,2,3};  
struct S3 {  
    struct S2 sinhvien;  
    float diem;  
} s3 = {{{"Cốc cốc", {4,5,6}}, 7};
```

A: S1 và S2 đúng B: S2 và S3 đúng C: S3 và S1 đúng D: Cả 3 cùng đúng

5. Đối với kiểu cấu trúc, cách gán nào dưới đây là không được phép:

A: Gán hai biến cho nhau.

B: Gán hai phần tử mảng (kiểu cấu trúc) cho nhau

C: Gán một phần tử mảng (kiểu cấu trúc) cho một biến và ngược lại

D: Gán hai mảng cấu trúc cùng số phần tử cho nhau

6. Cho đoạn chương trình sau:

```
struct {  
    int to ;  
    float soluong;  
} x[10];  
for (int i = 0; i < 10; i++) cin >> x[i].to >> x[i].soluong ;
```

Chọn câu đúng nhất trong các câu sau:

A: Đoạn chương trình trên có lỗi cú pháp

B: Không được phép sử dụng toán tử lấy địa chỉ đối với các thành phần to và soluong

C: Lấy địa chỉ thành phần soluong dẫn đến chương trình hoạt động không đúng đắn

D: Cả a, b, c đều sai

7. Chọn câu đúng nhất trong các câu sau:

A: Các thành phần của kiểu hợp (union) được cấp phát một vùng nhớ chung

B: Kích thước của kiểu hợp bằng kích thước của thành phần lớn nhất

C: Một biến kiểu hợp có thể được tổ chức để cho phép thay đổi được kiểu dữ liệu của biến trong quá trình chạy chương trình

D: a, b, c đúng

8. Cho khai báo:

```
union {  
    unsigned x;  
    unsigned char y[2];  
} z = {0xabcd};
```

Chọn câu đúng nhất trong các câu sau:

- A: Khai báo trên là sai vì thiếu tên kiểu
- B: Khởi tạo biến z là sai vì chỉ có một giá trị (0xabcd)
- C: `z.y[0] = 0xab`
- D: `z.y[1] = 0xab`

9. Cho kiểu hợp:

```
union U {  
    char x[1];  
    int y[2]; float z[3];  
} u;
```

Chọn câu đúng nhất trong các câu sau:

- A: `sizeof(U) = 1+2+3 = 6`
- B: `sizeof(U) = max(sizeof(char), sizeof(int), sizeof(float))`
- C: `sizeof(u) = max(sizeof(u.x), sizeof(u.y), sizeof(u.z))`
- D: b và c đúng

10. Cho khai báo:

```
union {  
    unsigned x;  
    struct {  
        unsigned char a, b;  
    } y;  
} z = {0xabcd};
```

Giá trị của `z.y.a` và `z.y.b` tương ứng:

- A: 0xab, 0xcd
- B: 0xcd, 0xab
- C: 0xabcd, 0
- D: 0, 0xabcd

11. Cho khai báo:

```
union {  
    struct {  
        unsigned char a, b;  
    } y;  
    unsigned x;
```

} z = {{1,2}};

Giá trị của z.x bằng:

A: 513

B: 258

C: Không xác định vì khởi tạo sai

D: Khởi tạo đúng nhưng z.x chưa có giá trị

12. Xét đoạn lệnh:

```
union U {  
    int x; char y;  
} u;  
u.x = 0; u.y = 200;
```

Tìm giá trị của u.x + u.y ?

A: 122

B: 144

C: 200

D: 400

13. Cho số phức dưới dạng cấu trúc gồm 2 thành phần là thực và ảo. Viết chương trình nhập 2 số phức và in ra tổng, tích, hiệu, thương của chúng.

14. Cho phân số dưới dạng cấu trúc gồm 2 thành phần là tử và mẫu. Viết chương trình nhập 2 phân số, in ra tổng, tích, hiệu, thương của chúng dưới dạng tối giản.

15. Tính số ngày đã qua kể từ đầu năm cho đến ngày hiện tại. Qui ước ngày được khai báo dưới dạng cấu trúc và để đơn giản một năm bất kỳ được tính 365 ngày và tháng bất kỳ có 30 ngày.

16. Nhập một ngày tháng năm dưới dạng cấu trúc. Tính chính xác (kể cả năm nhuận) số ngày đã qua kể từ ngày 1/1/1 cho đến ngày đó.

17. Tính khoảng cách giữa 2 ngày tháng bất kỳ.

18. Hiện thứ của một ngày bất kỳ nào đó, biết rằng ngày 1/1/1 là thứ hai.

19. Hiện thứ của một ngày bất kỳ nào đó, lấy ngày thứ hiện tại để làm chuẩn.

20. Viết chương trình nhập một mảng sinh viên, thông tin về mỗi sinh viên gồm họ tên và ngày sinh (kiểu cấu trúc). Sắp xếp mảng theo tuổi và in ra màn hình

21. Để biểu diễn số phức có thể sử dụng định nghĩa sau:

```
typedef struct { float re, im; } sophuc;
```

Cần bổ sung thêm trường nào vào cấu trúc để có thể lập được một danh sách liên kết các số phức.

22. Để tạo danh sách liên kết, theo bạn sinh viên nào dưới đây khai báo đúng cấu trúc tự trở sẽ được dùng:

Sinh viên 1: struct SV {char ht[25]; int tuoi; struct SV *tiep;};

Sinh viên 2: typedef struct SV node; struct SV {char ht[25]; int tuoi; node *tiep;};

Sinh viên 3: typedef struct SV {char ht[25]; int tuoi; struct SV *tiep;} node;

A: Sinh viên 1 B: Sinh viên 2 C: Sinh viên 2 và 3 D: Sinh viên 1, 2 và 3

23. Lập danh sách liên kết chứa bảng chữ cái A, B, C ... Hãy đảo phân đầu từ A .. M xuống cuối thành N, O, ... Z, A, ...M.
24. Viết chương trình tìm người cuối cùng trong trò chơi: 30 người xếp vòng tròn. Đếm vòng tròn (bắt đầu từ người số 1) cứ đến người thứ 7 thì người này bị loại ra khỏi vòng. Hỏi người còn lại cuối cùng ?
25. Giả sử có danh sách liên kết mà mỗi nốt của nó lưu một giá trị nguyên. Viết chương trình sắp xếp danh sách theo thứ tự giảm dần.
26. Giả sử có danh sách liên kết mà mỗi nốt của nó lưu một giá trị nguyên được sắp giảm dần. Viết chương trình cho phép chèn thêm một phần tử vào danh sách sao cho danh sách vẫn được sắp giảm dần.
27. Tạo danh sách liên kết các số thực x_1, x_2, \dots, x_n . Gọi m là trung bình cộng:
- $$m = \frac{x_1 + x_2 + \dots + x_n}{n}.$$

Hãy in lần lượt ra màn hình các giá trị: $m, x_1 - m, x_2 - m, \dots, x_n - m$.

28. Sử dụng kiểu union để in ra byte thấp, byte cao của một số nguyên.