Nguyễn Minh Đức
1712358

Phần trắc nghiệm

1. A

2. A

3. C

4. A

5. B (chương trình không có lệnh cout nên em nghĩ ý của cô ở đây là output i ra ngoài màn hình)

6. A

7. C

8. A

9. C

10. A (ở đây em mặc định chương trình nắm giữ cả node head và tail)

11. B

12. A B C (em thấy nối 2 danh sách trong cả 3 loại này đều có chi phí O(1), do mình đều nắm giữ node head và tail, chỉ việc nối tail của danh sách này với head danh sách kia)

13. D

14. B

15. D

Phần tự luận

1.

a. ++i: 1

b. ++(*pi): 11

c. --(*pd): 11.5

2.

a. int *p = 10;

lỗi p mang kiểu int* nên không thể khởi tạo bằng 1 biến kiểu int

b. int a = 10; int **p = &a;

lỗi p mang kiểu int** nên không thể khởi tạo bằng 1 biến kiểu int*, ở đây là &a

c. float x = 5; int **p = x;

tương tự ở đây p mang kiểu int** nên không thể khởi tạo bằng 1 biến kiểu float

3.

a. int *p; int **q; p = &q;

lỗi p là biến kiểu int* nên không thể được gán bằng 1 biến kiểu int***

b. int **p; int **q; p = &q;

lỗi p là biến kiểu int** nên không thể được gán bằng 1 biến kiểu int***

c. int **p; float *q; p = &q;

lỗi p là biến kiểu int** nên không thể được gán bằng 1 biến kiểu float**

4.

trong hàm fun() thì k chứa địa chỉ của j nên để j khi này là *k, nên muốn j chứa chỉ của a thì ta chỉ cần thêm lệnh

*k=a;

5.

*p++ và ++*p khác nhau, ở đây toán tử * với mức độ ưu tiên cao hơn sẽ được thực thi trước nên ta toán tử ++ sẽ tác động lên vùng nhớ mà p trỏ tới, ++ là hậu tố thì *p sẽ được lấy giá trị để thực thi phép toán trước rồi mới tăng *p lên 1. Còn ++ là tiền tố thì *p sẽ được tăng lên 1 trước rồi mới đưa vào phép toán.

6.

Khác nhau

+ char **a là con trỏ 2 cấp kiểu char, trỏ tới 1 con trỏ kiểu char. Khi khai báo ta có thể không cần khởi tạo giá trị trước.

+ char *a[] là 1 mảng các con trỏ kiểu char, khi khởi tạo thì giá trị a chính là địa chỉ đầu tiên của ô nhớ trong mảng, nhưng a này không có ô nhớ riêng như kiểu char**, và các ô nhớ này là ô nhớ tĩnh, nên ta không thể thay đổi giá trị của a, đồng thời khi khởi tạo ta cần định trước giá trị cho số ô nhớ.

+ char a[][] là mảng 2 chiều kiểu char, gần giống char *a[], về vấn đề giá trị của a, nhưng các giá trị trong mảng a đều là kiểu char, và số ô nhớ của a[m][n] là cố định m.n, trong khi với char *a[] thì mỗi phần tử trong a lại có thể trỏ tới 1 mảng char riêng chứa số phần tử khác nhau.