C Programming

Tutorial 4

1. Discuss the difference between the following declarations and the use of them.
    a. struct astruct { … };

    b. typedef struct { … } astruct;

    c. typedef struct astruct { … } astruct;

2. Do you see any problems with the following type definition?

```
typedef struct {
        char *data;
        ptr_node next;
} *ptr_node;
```

3. int func(); will return an integer. If I want func() returns an int and c char, can you do that? How many solutions can you give?

```
typedef struct {
    int i;
    char c;
} int_char;
```

4. See the following struct type:

```
typedef struct {
    int i;
    float f;
    } int_float;
int_float a, b, c;
… /* a and b are defined */
c = func(a, b);
```

Can you write a function func(), which do c = a + b?

5. See the following declarations:

```
typedef struct {
    int i;
    char c;
    int ii;
    } int_char_int;

typedef struct {
    int i;
    char c;
    char cc;
    int ii;
    } int_char_char_int;
```

Can you tell the size of the two user-defined types on a 32-bit machine?

6. Discuss the difference between Structure and Union?

7. See the following enumeration declaration:

```
typedef enum day {mon, tue, wed, … } day;
day today;
…
```

Can you print out “mon”, “tue”, “wed”, …, using the integral values behind them?

8. Do you see any problems from the following statements?

```
char a1[] = "ABCD", a2[] = "abcdef", a3[] = "12345";
strcpy(a1, a2);
strcat(a2, a3);
```