柱

6.1

a)陣列b)索引c符號常數 D)排序e)搜尋 F)二維

6.2

a)非。陣列只能存放相同型別的數值。

b)非。索引必須是整數或是整數運算式。

c)非。C語言會自動將剩下的元素初始化為零。

d)是。

e)非。個別的陣列元素是以傳值呼叫傳遞給函式的。假如是整個陣列傳遞給函式,則任何元素的修改都會影響到原來的陣列。

6.3

a) #defineSIZE10

b) doublefractions[SIZE] = {0.0};

c) fractions[4]

d) fractions [9] =1.667;

e) fractions[6] =3.333;

f) printf("%.2f %.2f\n", fractions[6], fractions[9]);

輸出:3.33 1.67

for ( x = 0; x<SIZE; ++x )

printf( "fraction [%u] = %f\n", x, fractions [x]);

輸出:

fractions[0] =0.000000

fractions[1] = 0.000000

fractions [2] = 0.000000

fractions [3] = 0.000000

fractions [4] = 0.000000

fractions [5] = 0.000000

fractions [6] =3.333000

fractions [7] = 0.000000

fractions [8] =0.000000

fractions [9] = 1.667000

6.4

a) int table[SIZE] [SIZE];

b) Nine elements. printf("%d\n", SIZE \* SIZE);

c) for (x = 0; x <SIZE; ++x) {

for (y = 0; y <SIZE; ++y) {

table [x] [y] = x + y;

d) for (x = 0; x <SIZE; ++x) {

for (y = 0; y <SIZE; ++y) {

printf("table[%d] [%d] = %d\n", x, y, table[x] [y]);

}

}

輸出:

table [0] [0] = 1

table [0] [1] = 8

table [0] [2] = 0

table [1] [0] = 2

table [1] [1] = 4

table [1] [2] = 6

table [2] [0] = 5

table [2] [1] = 0

table [2] [2] = 0

6.5

a)錯 : #define前置理器命令之後多了個分號。

更正:去掉分號。

b)錯#:指定敘述式將值設給符號常數

更正:在#define 前置處理器命令當中為符號常數設值,例如#define SIZE10,不須使用指定運算子

c)錯誤:引用到一個超出陣列邊界的元素(b[10])。

更正:將控制變數的終值改為9。

d)錯:#include前置處理器命令之後多了個分號。

更正:去掉分號。

E)錯誤:陣列的索引表示法錯誤。

更正:將敘述式改為a[1][1]=5

F)錯誤:用指定敘述式將值設給符號常數。

更正:在#define 前置處理器命令中為符號常數設值,例如#defineVALU120,不需使

用指定運算子。