

## 錬成問題

■ `int`型の変数`n1`, `n2`, `n3`の値が、それぞれ21, 6, 0であるとする。このとき、以下の各式を評価した値を示せ。

<code>n1, n2</code> … <input style="width: 60px;" type="text" value="(1)"/>	<code>n2, n3</code> … <input style="width: 60px;" type="text" value="(2)"/>
<code>n1++, n2++</code> … <input style="width: 60px;" type="text" value="(3)"/>	<code>n2, 1</code> … <input style="width: 60px;" type="text" value="(4)"/>
<code>n1 = (n3 == 0)</code> … <input style="width: 60px;" type="text" value="(5)"/>	<code>n2 = n3++</code> … <input style="width: 60px;" type="text" value="(6)"/>
<code>n1 = n2 = 0</code> … <input style="width: 60px;" type="text" value="(7)"/>	<code>n1 = (n2, n3)</code> … <input style="width: 60px;" type="text" value="(8)"/>
<code>2 * (n1 = n2)</code> … <input style="width: 60px;" type="text" value="(9)"/>	<code>n1 / (n2, ++n3)</code> … <input style="width: 60px;" type="text" value="(10)"/>

■ 配列の各要素は、配列の名前に続けて演算子と呼ばれる`[ ]`の中にと呼ばれる整数値を入れることによって表す。

なお、配列の各要素の型は、。

▶ の選択肢 … (a) 全て同じである (b) 要素によって異なってもよい

■ 要素数が5で要素型が`int`型である配列`x`の各要素を先頭要素から順に10, 20, 30, 40, 50で初期化するのであれば、以下のように宣言する。

```
int x[5] = ;
```

このとき、`[]`中の5は。

▶ の選択肢 … (a) 省略できない (b) 省略してもよい

■ 以下に示すのは、いずれも、要素数が5で要素型が`int`型である配列`a`の各要素に対して、先頭から順に1, 3, 5, 7, 9を代入するプログラム部分である。

```
for (i = 0; ; i++)
    a[i] = ;
```

```
for (i = 4; ; i--)
    a[i] = ;
```

```
i = 5;
while ( > 0)
    a[i] = ;
```

```
i = 4;
while ( > 0)
    a[i] = ;
```

■ 右に示すのは、要素数が`no`で要素型が`int`型の配列`a`の要素の最大値を求めるプログラム部分である。

```
max = ;
for (; i < no; i++)
    if (max < a[i])
        max = ;
```

■ 右に示すのは、要素数が`no`で要素型が`int`型の配列`a`の要素の最小値を求めるプログラム部分である。

```
min = ;
for (; i >= 0; i--)
    if (min > a[i])
        min = ;
```

■ 右に示すプログラム部分の実行結果を示せ。

```
a[0] = (30)
a[1] = (31)
a[2] = (32)
a[3] = (33)
```

```
int i;
int a[4] = {10, 20};

for (i = 0; i < 4; i++)
    printf("a[%d] = %d\n", i, a[i]);
```

■ 右に示すのは、要素数が  $no$  で要素型が `int` 型である配列  $x$  の要素を、配列  $y$  の対応する (添字が同じ) 要素にコピーするプログラム部分である。

```
for ((34); i < no; i++)
    y[i] = (35);
```

■ 右に示すのは、要素数が  $no$  で要素型が `int` 型である配列  $x$  の要素を、配列  $y$  の要素に逆順にコピーするプログラム部分である。

```
for ((36); i < no; i++)
    (37) = x[i];
```

■ 右に示すのは、要素数が11で要素型が `double` 型である配列  $x$  に、1.0, 1.1, ..., 2.0 を代入するプログラム部分である。

```
for (i = 0; (38); i++)
    x[i] = (39);
```

■ 右に示すのは、要素数が  $no$  で要素型が `double` 型である配列  $x$  の全要素の合計値を求めるプログラム部分である。

```
double sum = (40);
while ((41))
    (42) += x[no];
```

■ 右に示すのは、要素数が  $no$  で要素型が `double` 型である配列  $x$  のうち、値が正である要素の個数を求めるプログラム部分である。

```
int cnt = (43);
while ((44))
    if ((45)) cnt++;
```

■ 右に示すのは、要素数が  $no$  で要素型が `double` 型である配列  $x$  のうち、0以上である要素の添字と値を先頭から順に表示するプログラム部分である。

```
for (i = 0; (46); i++) {
    if ((47)) {
        printf("添字 = %d ", i);
        printf("値 = %f\n", (48));
    }
}
```

■ 右に示すプログラム部分の誤りを全て指摘せよ。

```
(49)
```

```
int i;
int ax[5] = {0, 1, 2, 3, 4, 5};

for (i = 0; i <= 6; i++)
    printf("ax[%d] = %d\n", i, ax[i]);
```

(次ページへ続く)

- 右に示すプログラム部分の実行結果を示せ。

```
(50)
```

```
int i, j;
for (i = 0, j = 7; i < 5; i++, j--)
    printf("%d%d", i, j);
```

- 以下に示すのは、いずれも、要素数が `no` で要素型が `int` 型である配列 `a` の全要素に対して整数値 `x` を代入するプログラム部分である。

```
for (i = 0; (51); i++)
    a[i] = x;
```

```
i = no;
while ((52) > 0)
    a[i] = x;
```

- 以下に示すのは、要素数が `no` で要素型が `double` 型である配列 `vx` の全要素の並びを逆順に並べかえるプログラム部分である。

```
for (i = 0; i < (53); i++) {
    (54) temp = vx[i];
    vx[i] = vx[no - i - 1];
    (55) = temp;
}
```

- 以下に示すのは、`NUMBER` 人の点数を配列 `tensu` に読み込み、優 (80～100点)、良 (70～79点)、可 (60～69点)、不可 (0～59点)、判定不能 (0未満あるいは100点より上) の人数を、それぞれ `a`, `b`, `c`, `d`, `e` に格納するプログラム部分である。

```
#define NUMBER 10

int i;
int tensu[NUMBER];          /* 5人の学生の点数 */
int a=0, b=0, c=0, d=0, e=0; /* 優・良・可・不可・判定不能の人数 */

puts("点数を入力してください。");
for (i = 0; i < NUMBER; i++) {
    printf("%2d番:", i + 1);
    scanf("%d", (56));
    if ((57) || (58)) /* 0～100以外 */
        e++;
    else if ((59)) /* 優: 80～100 */
        a++;
    else if ((60)) /* 良: 70～79 */
        b++;
    else if ((61)) /* 可: 60～69 */
        c++;
    else /* 不可: 60未満 */
        d++;
}
```

- 上問における `NUMBER` は、(62) 形式マクロと呼ばれる。

■ 右に示すプログラム部分の実行結果を示せ。

(63)

```
int a = 0;
int b = 1;
putchar((a = b) ? '+' : '-');
```

■ 右に示すプログラム部分の実行結果を示せ。

(64)

```
int a = 0;
int b = 0;
putchar((a = b) ? '+' : '-');
```

■ 右に示すのは、要素数が $no$ で要素型が $int$ 型である配列 $a$ の要素の値の幅、すなわち最大値と最小値の差を求めて $range$ に代入するプログラム部分である。

```
min = (65) = (66);
for ((67); i < no; i++) {
    if (max < a[i]) (68) = a[i];
    if (min > a[i]) (69) = a[i];
}
range = (70) - (71);
```

■ 右に示すのは、要素数が $no$ で要素型が $int$ 型である配列 $x$ のうち、0より大きい要素の添字を全て配列 $c$ に先頭から順に、その個数を $d$ に格納するプログラム部分である。

```
d = (72);
for (i = 0; (73); i++)
    if ((74))
        c[(75)] = i;
```

※ たとえば、 $x$ の要素が1,0,2,-1,3,2であれば、配列 $c$ には0,2,4,5を先頭から順に代入し、 $d$ には4を代入する。

■ 以下に示すのは、2行3列の行列を表す2次元配列 $mx$ 、 $my$ の差を $mz$ に格納して表示するプログラムである。

```
#include (76)

int main(void)
{
    int (77);
    int mx[2][3] = { {1, 2, 3}, {4, 5, 6} };
    int my[2][3] = { {6, 3, 4}, {5, 1, 2} };
    int mz[2][3];

    for (i = 0; i < 2; i++)
        for (j = 0; j < 3; j++)
            mz[i][j] = (78) - (79);

    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++)
            printf("%3d", (80));
        (81)('\n');
    }

    (82) (0);
}
```

```
-5 -1 -1
-1 4 4
```

(次ページへ続く)

- 右に示すプログラム部分の実行結果を示せ。

```

a = 
b = 
c = 
d = 

```

```

double a;
int b, c, d;
a = b = 1.5;
c = d = 1;
printf("a = %f\n", a);
printf("b = %d\n", b);
printf("c = %d\n", c);
printf("d = %d\n", d);

```

- 右に示すプログラム部分の実行結果を示せ。

```

a[0][0] = 
a[0][1] = 
a[1][0] = 
a[1][1] = 

```

```

int a[2][2] = {0};
for (i = 0; i < 2; i++)
    for (j = 0; j < 2; j++)
        printf("a[%d][%d] = %d\n",
               i, j, a[i][j]);

```

- 以下に示すのは、2行3列の行列を表す2次元配列 *mx*, *my* の各要素の値を読み込み、和を *mz* に格納して表示するプログラムである。

```

 <stdio.h>

int main(void)
{
    int ;
    int mx[2][3], my[2][3], mz[2][3];

    for (i = 0; i < 2; i++)
        for (j = 0; j < 3; j++) {
            printf("", i, j);
            scanf("%d", );
        }

    for (i = 0; i < 2; i++)
        for (j = 0; j < 3; j++) {
            printf("", i, j);
            scanf("%d", );
        }

    for (i = 0; i < 2; i++)
        for (j = 0; j < 3; j++)
            mz[i][j] =  + ;

    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++)
            printf("%3d", );
        putchar();
    }

    return (0);
}

```

```

mx[0][0] : 
mx[0][1] : 
mx[0][2] : 
mx[1][0] : 
mx[1][1] : 
mx[1][2] : 
my[0][0] : 
my[0][1] : 
my[0][2] : 
my[1][0] : 
my[1][1] : 
my[1][2] : 
7 5 7
9 6 8

```

■ 右に示すのは、2次元配列  $m$  を、 $no$  行  $no$  列の零行列を表すようにするプログラム部分である。

※ 全ての要素に 0 を代入する。

```
for (i = 0; i < no; i++)
    for (j = 0; j < no; j++)
        m[i][j] = (101);
```

■ 右に示すのは、2次元配列  $m$  を、 $no$  行  $no$  列の単位行列を表すようにするプログラム部分である。

※ 対角要素には 1 を、それ以外には 0 を代入する。

```
for (i = 0; i < no; i++)
    for (j = 0; j < no; j++)
        m[i][j] = (102);
```

■ 以下に示すのは、3行4列の行列  $a$  と、4行3列の行列  $b$  の積を、3行3列の行列  $c$  に格納し、その結果を表示するプログラム部分である。

```
for (i = 0; i < 3; i++) {
    for (j = 0; (103); j++) {
        c[i][j] = (104);
        for (k = 0; (105); k++)
            c[i][j] += a[i][(106)] * b[(107)][j];
    }
}

for (i = 0; i < 3; i++) {
    for (j = 0; j < (108); j++)
        printf("c[%d][%d] = %d\n", i, j, (109));
}
```

■ 上問の配列  $a$ ,  $b$ ,  $c$  は2次元配列であるが、配列の次元数は(110)。

▶ (110)の選択肢 … (a) 高々2である (b) 2以上でもよい

■ 以下に示すプログラムの誤りを全て指摘せよ。

(111)

```
$include <stdio>

int menu(void)
{
    int no = 5;
    int ary[no];

    for (i = 0; i <= no; i++)
        printf("a[%d] = ", i);          /* 整数値の入力を促す */
        scanf("%d", ary[i]);           /* 整数値を読み込む */

    for (i = 0; i <= no; i++)
        printf("ary[%d] = %d\n", i, ary[i]);

    return (0);
}
```