プログラミング及び実習2

第 6 回課題（ex06）の解答例

佐野 彰

ex06 解答例: mult35.c（配列要素のうち3と5の倍数）

```c
#include <stdio.h>

#define N 99

int main() {
    int n;
    int a[N];

    scanf("%d", &n);
    if (n > N || n <= 0) return 0;

    for (int i=0; i<n; i++)
        scanf("%d", &a[i]);

    for (int i=0; i<n; i++)
        if (a[i]%3 == 0)
            printf("%d ", a[i]);
    printf("\n");

    for (int i=0; i<n; i++)
        if (a[i]%5 == 0)
            printf("%d ", a[i]);
    printf("\n");

    return 0;
}
```

- 先に n 個をすべて配列に読み込み、その配列を 2 回走査する
- 倍数が 1 つもない行も、`printf("\n");` で空行を出す
- 負の数も `a[i]%3 == 0` で判定できる（`-9 % 3` は 0）

ex06 解答例: asterisk.c, asterisk2.c（アスタリスク）

```c
// asterisk.c
#include <stdio.h>

int main() {
    int n;
    do {
        scanf("%d", &n);
    } while (n < 1);

    for (int i=0; i<n; i++) {
        for (int j=0; j<=i; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

```c
// asterisk2.c
#include <stdio.h>

int main() {
    int n;
    do {
        scanf("%d", &n);
    } while (n < 1);

    for (int i=0; i<n; i++) {
        for (int j=0; j<i; j++)
            printf(".");
        for (int j=0; j<n-i; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

- 外側のループが行、内側のループがその行の文字を担当する
- i を 0 から数えるので、i 行目の `*` は i+1 個（`j<=i`）、`.` は i 個になる

ex06 解答例: maxmin.c（最大値・最小値・平均値）

```c
#include <stdio.h>

int main() {
    int n;
    do {
        scanf("%d", &n);
    } while (n < 1);

    int max, min, sum = 0;
    for (int i=0; i<n; i++) {
        int a;
        scanf("%d", &a);
        if (i == 0 || a > max) max = a;
        if (i == 0 || a < min) min = a;
        sum += a;
    }
    printf("%d %d %.3f\n", max, min, (double)sum/n);

    return 0;
}
```

- 最初の値（`i == 0`）で `max`・`min` を初期化する。`max` を `0` で初期化すると、入力がすべて負のときに誤る
- `sum/n` は整数の割り算になる。`(double)sum/n` のように型変換してから割ること

ex06 解答例: select.c（昇順の並び替え）

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n < 1) return 0;

    int a[n];
    for (int i=0; i<n; i++)
        scanf("%d", &a[i]);
    for (int k=n-1; k>0; k--) {
        int m = 0;
        for (int i=1; i<=k; i++) {
            if (a[i] > a[m])
                m = i;
        }
        int tmp = a[m];
        a[m] = a[k];
        a[k] = tmp;
    }

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

- `a[0]` 〜 `a[k]` の最大値の位置 `m` を探し、内側のループが終わってから `a[m]` と `a[k]` を 1 回だけ入れ替える
- 最大値が後ろから順に確定する（`a[n-1]`、`a[n-2]`、…）ので、昇順に並ぶ
- 2 つの値の交換には、一時変数 `tmp` が必要（`a[m] = a[k];` `a[k] = a[m];` では元の値が消える）

別解（見つけるたびに入れ替える）

```c
    for (int i=0; i<n-1; i++) {
        for (int j=i+1; j<n; j++) {
            if (a[i] > a[j]) {
                int tmp = a[i];
                a[i] = a[j];
                a[j] = tmp;
            }
        }
    }
```

- スライド 06 の最初の形。結果は同じだが入れ替えの回数が多い。位置だけ覚えて最後に 1 回入れ替えるのが本来の選択ソート

ex06 解答例: hist10.c（要素の数え上げ）

```c
#include <stdio.h>
#define N 11

int main() {
    int n;
    scanf("%d", &n);
    if (n < 1) return 0;

    int a[N];
    for (int i=0; i<N; i++) a[i] = 0;

    int val;
    for (int i=0; i<n; i++) {
        do {
            scanf("%d", &val);
        } while (val < 0 || val > 100);
        a[val/10]++;
    }

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

    return 0;
}
```

- 区間は [0,9] 〜 [90,99] と [100] の 11 個なので、配列の大きさは 11
- `val/10` が区間の番号になる（例: 49/10 は 4、100/10 は 10）
- カウント用の配列は、使う前に 0 で初期化すること
- 範囲外の値は `do` 〜 `while` で読み飛ばす。読み飛ばした値は n 個に数えない

ex06 解答例: kukudrop.c（歯抜けの掛け算九九表）

```c
#include <stdio.h>

int main() {
    int n, k;
    scanf("%d%d", &n, &k);
    if (n < 1 || n > 9) return 0;

    printf("    ");
    for (int i=1; i<=n; i++)
        printf("%3d", i);
    printf("\n\n");

    for (int i=1; i<=n; i++) {
        printf("%3d ", i);
        for (int j=1; j<=n; j++) {
            int p = i*j;
            if (p%10 == k || (p/10 == k && p > 9))
                printf("   ");
            else
                printf("%3d", p);
        }
        printf("\n");
    }
    return 0;
}
```

- 積は最大 81 なので、桁は一の位 `p%10` と十の位 `p/10` の 2 つだけ調べればよい
- `p > 9` がないと、k が 0 のとき 1 桁の積（十の位が 0）まで空白になる
- k が 0〜9 以外なら、どちらの条件も成り立たず、表がそのまま出る
- 空白も `"   "`（3 文字）で出し、列をそろえる

ex06 解答例: maxminnum.c（最大値・最小値の要素番号）

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n < 1) return 0;

    int max, min;
    int maxi, mini;
    for (int i=0; i<n; i++) {
        int a;
        scanf("%d", &a);
        if (i == 0 || a >= max) {max = a; maxi = i;}
        if (i == 0 || a <= min) {min = a; mini = i;}
    }
    printf("%d %d\n", max, maxi);
    printf("%d %d\n", min, mini);

    return 0;
}
```

- 同じ値が複数あるときは最後の番号を出すので、比較は `>=` ・ `<=` にする（`>` だと最初の番号になる）

ex06 解答例: maxminnum2.c（最大値・最小値の要素番号2）

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n < 1) return 0;

    int a[n];
    int max, min;
    for (int i=0; i<n; i++) {
        scanf("%d", &a[i]);
        if (i == 0 || a[i] > max) max = a[i];
        if (i == 0 || a[i] < min) min = a[i];
    }
```

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

    printf("%d", min);
    for (int i=0; i<n; i++)
        if (a[i] == min) printf(" %d", i);
    printf("\n");

    return 0;
}
```

- 最大値が決まるのは全部読んだ後なので、値を配列に残しておく
- 1 回目のループで最大値・最小値を求め、2 回目のループで等しい要素の番号を順に出す