

## アルゴリズムとデータ構造 授業中練習問題11

次のプログラムは「構造体の単純交換ソートの実現例」(教科書の List6-1 を構造体に拡張)である。このプログラムに関して、以下の問いに答えなさい。さらに、このプログラムを入力し、自分のパソコンでコンパイル、実行できることを確認してください。なお、プログラムの日本語部分は、英語、ローマ字に変更してかまいません。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define swap(type, x, y) do {type t; t = x; x = y; y = t;} while(0)

#define ASCENDING      0 /* 昇順 */
#define DESCENDING     1 /* 降順 */

/*--- 会員データ ---*/
typedef struct {
    int no;          /* 番号 */
    char name[20];   /* 氏名 */
} Member;

/*--- 会員の番号の比較関数 ---*/
int MemberNoCmp(const Member *x, const Member *y) {
    return x->no < y->no ? -1 : x->no > y->no ? 1 : 0;
}

/*--- 会員の氏名の比較関数 ---*/
int MemberNameCmp(const Member *x, const Member *y) {
    return strcmp(x->name, y->name);
}

/*--- 会員データ（番号と氏名）の表示（改行あり） ---*/
void PrintLnMember(const Member *x) {
    printf("%d %s¥n", x->no, x->name);
}

/* --- 単純交換ソート --- */
void bubble(Member *a, int n,
            int compare(const Member *x, const Member *y),
            int order) {
    int i, j;

    for(i = 0; i < n - 1; i++) {
        for(j = n-1; j > i; j--)
            if (compare(a + j - 1 + order, a + j - order) > 0)
                swap(Member, a[j-1], a[j]);
    }
}

/*--- 全データの表示 ---*/
void Print(const Member *data, int n) {
    int i;
```

```

for(i=0; i < n; i++)
    PrintLnMember(data+i);
}

/*--- メニュー ---*/
typedef enum {
    TERMINATE, ASCEND_NO, ASCEND_NAME,
    DESCEND_NO, DESCEND_NAME, PRINT_ALL
} Menu;

/*--- メニュー選択 ---*/
Menu SelectMenu(void) {
    int i, ch;
    char *mstring[] = {
        "番号で昇順ソート", "名前で昇順ソート",
        "番号で降順ソート", "名前で降順ソート",
        "データを表示"
    };

    do {
        for (i = TERMINATE; i < PRINT_ALL; i++) {
            printf("(%2d) %-24.24s ", i + 1, mstring[i]);
            if ((i % 3) == 2)
                putchar('\n');
        }
        printf("( 0) 終了 :");
        scanf("%d", &ch);
    } while (ch < TERMINATE || ch > PRINT_ALL);

    return (Menu)ch;
}

/*--- メイン ---*/
int main(void) {
    Menu menu;
    Member data[] = {
        {5, "umeda"}, {7, "satoshi"},
        {6, "noyuri"}, {0, "daisuke"},
        {0, "motoko"}, {4, "agemi"},
        {9, "ito"}, {2, "ohta"},
        {1, "takashi"}, {3, "kouji"}
    };
    int ndata = sizeof(data)/sizeof(data[0]);

    do {
        int n;

        switch (menu = SelectMenu()) {
            case ASCEND_NO : /* 番号で昇順にソート */
                bubble(data, ndata, MemberNoCmp, ASCENDING);
                break;
            case ASCEND_NAME : /* 名前で昇順にソート */

```

```

        bubble(data, ndata, MemberNameCmp, ASCENDING);
        break;
    case DESCEND_NO : /* 番号で降順にソート */
        bubble(data, ndata, MemberNoCmp, DESCENDING);
        break;
    case DESCEND_NAME : /* 名前で降順にソート */
        bubble(data, ndata, MemberNameCmp, DESCENDING);
        break;
    case PRINT_ALL : /* 全データを表示 */
        Print(data, ndata);
        break;
    }
} while (menu != TERMINATE);

return 0;
}

```

- 1) このプログラムの動作直後に、数字の 5 を入力したときの data->name が指している文字を示しなさい。
- 2) このプログラムの動作直後に、数字の 3 を入力したときの data->name が指している文字列を示しなさい。
- 3) このプログラムの動作直後に、数字の 3 を入力したときの bubble 関数中の compare(a+j-1+order, a+j-order)が呼び出される回数は何回ですか。
- 4) このプログラムの動作直後に、数字の 3 を入力したときの bubble 関数中で i=0 のとき、swap(Member , a[j-1], a[j]) が呼び出される回数は何回ですか。
- 5) このプログラムの動作直後に、数字の 4 を入力したときの data->name が指している文字列を示しなさい。
- 6) 5)の状態から更に、数字の 2 を入力したときの data->name が指している文字列を示しなさい。
- 7) 5)の状態から更に、数字の 2 を入力したときの bubble 関数中の compare(a+j-1+order, a+j-order) が呼び出される回数は何回ですか。
- 8) 5)の状態から更に、数字の 2 を入力したときの bubble 関数中で i=0 のとき、swap(Member , a[j-1], a[j])が呼び出される回数は何回ですか。