

```
// 「Cで学ぶデータ構造とアルゴリズム」(西原清一) オーム社, 2008
// 図2・12 (p.39) 順配置されたスタックの操作
```

```
#include <stdio.h>
#define M 5
#define OVERFLOW -1
#define UNDERFLOW -2

int x[M], y, t;

int push()
{   if (t < M) x[t++] = y;
    else return OVERFLOW;
    return 0;
}

int pop()
{   if (t > 0) y = x[--t];
    else return UNDERFLOW;
    return 0;
}

void print_stack()
{   int i;
    printf(" *STACK at present(t=%d) is = (", t);
    for (i=0; i<t;) printf(" %d", x[i++]);
    printf(" )\n\n");
}

main()
{   int i;

    t = 0;
    printf("add:positive datum, remove:a negative datum, finish:0 OK?");
    printf(" Let's go.\n\n");

    for(;;) {
        printf(">");
        scanf("%d",&y);
        if (y > 0)
```

```

        if ((i = push()) >= 0) print_stack();
        else printf(" **stack overflow(=%d)¥n", i);    //overflow
    else
    if (y < 0)
        if ((i = pop()) >= 0)
            printf(" *popped data is %d¥n", y);
        else printf(" **stack underflow(=%d)¥n", i);    //underflow
    else
        if (y == 0) {print_stack(); break;}
    }
}

```