```
// 「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・13 (p.39) 順配置された待ち行列の操作

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

int x[M],y,f,r;

int enqueue()
{   if (f-r == 1 || f-r+M == 1) return OVERFLOW;
    x[r] = y;
    if (++r == M) r = 0;
    return 0;
}

int dequeue()
{   if (f == r) return UNDERFLOW;
    y = x[f++];
    if (f == M) f = 0;
    return 0;
}

void print_queue()
{   int i;
    printf("  *QUEUE at present(f=%d, r=%d) is = (",f,r);
    for (i=0; i<M;) printf(" %d",x[i++]);
    printf(" )¥n¥n");
}

main()
{   int i;

    f = 0;  r = 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 = enqueue()) >= 0) print_queue();
            else printf("    **queue overflow(=%d) at f=%d, r=%d¥n",i,f,r);
//overflow
        else
        if (y < 0)
            if ((i = dequeue()) >= 0)
                printf(" *dequeued data is %d¥n",y);
            else printf("    **queue underflow(=%d) at f=%d, r=%d¥n",i,f,r);
//underflow
        else
            if (y == 0) {print_queue(); break;}
    }
}
```