

```
// 「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
//  解図2・A (p.207-208) リンク配置で実現した待ち行列
```

```
#include <stdio.h>
#include <stdlib.h>
#define  EMPTY -9

struct queue
{
    int data;
    struct queue *next;
};

struct queue *front=NULL,*rear=NULL,*p;

void get_cell()
{
    p =(struct queue *) malloc(sizeof(struct queue));
    if (p == NULL) {
        printf("malloc error\n");
        exit(EXIT_FAILURE);
    }
}

void enqueue(int value)
{
    get_cell();
    p->data = value;
    p->next = NULL;
    if (front == NULL)
        front = p;
    else
        rear->next = p;
    rear = p;
}

int dequeue()
{
    int value;
    if (front == NULL) return EMPTY;
    value = front->data;
    p = front;
    front = front->next;
    free(p);
}
```

```

        return value;
    }

main()
{
    int i;
    printf("input data(>0:enqueue, -1:dequeue, 0:stop\n");
    for (;;) {
        printf("> ");
        scanf("%d",&i);
        if (i == 0) break;
        if (i > 0) enqueue(i);
        else {
            i = dequeue();
            if (i == EMPTY) printf("queue is empty!\n");
            else printf("dequeued data = %d\n",i);
        }
    }
    p = front;
    while (p != NULL) {
        printf("%d => ",p->data);
        p = p->next;
    }
    printf(" REAR\n\n",front,rear);
}

```