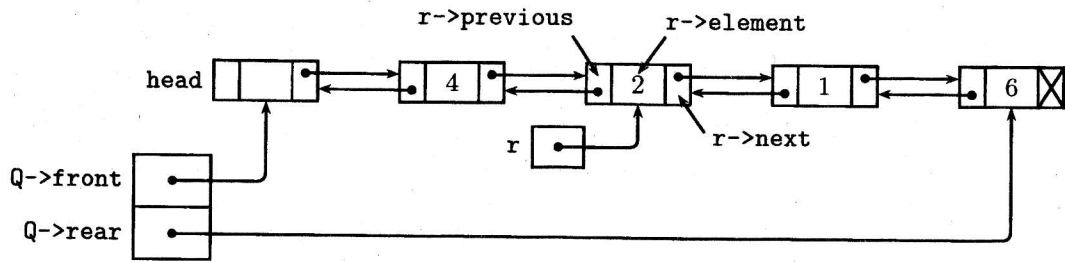


【問3】 待ち行列（キュー）の変形として、先頭と末尾の両端への挿入と削除を行うことができる両端キュー（double-ended queue）というデータ構造がある。両端キューは、以下の図のように、双方向リスト（直前と直後の両方のセルを指している線形リスト）を用いて効率良く実現することができる。



このように実現される両端キューの先頭にデータを挿入する関数 `insert_front` と先頭のデータを削除する関数 `delete_front`（および構造体 `struct cell`）の定義を以下に示す。これらのプログラムの空欄 (1) ~ (4) に入るべき適切な記述を、選択肢 (a) ~ (y) の中から選べ。(20 点)

```

struct cell {
    int element;
    struct cell *next, *previous;
};

void insert_front(int x, struct queue *Q)
{
    struct cell *p;

    p = (struct cell *)malloc(sizeof(struct cell));
    p->element = x;
    p->next = (1);
    p->previous = Q->front;
    Q->front->next = p; /* head が p を指すようにする */
    if(p->next == NULL) /* Q が空の場合 */
        Q->rear = (2);
    else /* Q が空でない場合 */
        p->next->previous = (2);
}

void delete_front(struct queue *Q)
{
    struct cell *p;

    if(Q->front == Q->rear) printf("キューが空なので、削除できません!\n");
    else {
        p = Q->front->next;
        Q->front->next = Q->front->next->next;
        free(p);
        if(Q->front->next == NULL) /* Q が空になった場合 */
            Q->rear = Q->front;
        else /* Q が空でない場合 */
            (3) = (4);
    }
}

```