```
    struct node *temp;
    //łåforå
    for (p = head->next; p != NULL; p = p->next)
    {
        if (p->next != NULL && key == p->next->key)
        {//pkeyåå£åå
            //åŁ¢åŁ
            temp = p->next->next;
            //keyåå¢ł
            free(p->next);
            //pnextåŁ¢
            p->next = temp;
            //åŁ1
            return 1;
        }
    }
    //keyååł0
    return 0;
}
0 -> 1 -> 3 -> 4 -> 5 -> 8 -> 10 -> 12 ->
0 -> 1 -> 3 -> 4 -> 8 -> 10 -> 12 ->
/**
 * @brief łkeyåłåłåłåłåå
 * @param *head:ååłå¢
 *        *oddhead:åłå¢
 *        *evenhead:åłå¢
 */
void oddeven(struct node *head, struct node *oddhead, struct node *evenhead)
{
    //å
    struct node *oddlist = oddhead, *evenlist = evenhead, *p;
    //headåforå
    for (p = head->next; p != NULL; p = p->next)
    {
        //ååå
        if ((p->key) % 2 != 0)
        {
            //åååłp¢åŁ
            oddlist->next = p;
            //oddlistå
            oddlist = oddlist->next;
        }
        else
        {
            //ååå
            evenlist->next = p;
            evenlist = evenlist->next;
        }
    }
    //łåNULLåŁ
    oddlist->next = NULL;
    evenlist->next = NULL;
}
0 -> 1 -> 3 -> 4 -> 5 -> 8 -> 10 -> 12 ->
0 -> 4 -> 8 -> 10 -> 12 ->
1 -> 3 -> 5 ->
```