Node * ziplist(Node * a, Node * b)
// pre: 0 < size(a) and 0 < size(b) and size(b) < size(a)
// post: returns pointer to first node of list created by
// zipping b into a (pointers/lists a and b are changed)
{
    bool left = true;
    Node * pointer = a;
    Node * temp = a;

    while (b->next != 0)
    {
        if (left == true)
        {
            temp = a;
            a->next = b;
            left = false;
            a = temp->next;
        }
        else if (left == false)
        {
            temp = b;
            b->next = a;
            left = true;
            b = temp->next;
        }
    }

    return pointer;
}
Node * ziplist(Node * a, Node * b)
// pre: 0 < size(a) and 0 < size(b) and size(b) < size(a)
// post: returns pointer to first node of list created by
// zipping b into a (pointers/lists a and b are changed)
{
    Node * temp;
    Node * aList = a;
    Node * bList = b;
    Node * aListNext, bListNext;

    while (bList != a) {  
        aListNext = aList -> next;    // these store next nodes in a, b
        bListNext = bList -> next;
        temp = aList -> next;        // temp points to next element of a
        aList -> next = bList;       // a->next points to b
        bList -> next = temp;        // b->next points to next element of a
        aList = aListNext;
        bList = bListNext;          // go to next element of a, b
    }

    return a;
}
Node * ziplist(Node * a, Node * b)
// pre: 0 < size(a) and 0 < size(b) and size(b) < size(a)
// post: returns pointer to first node of list created by
// zipping b into a (pointers/lists a and b are changed)
{
    Node * start = a;
    Node * nextb = 0;
    do {
        nextb = b->next;
        b->next = a->next;
        a->next = b;
        a = b = nextb;
    } while (b != 0);
    return start;
}
Node * ziplist(Node * a, Node * b)
// pre: 0 < size(a) and 0 < size(b) and size(a) < size(b)
// post: returns pointer to first node of list created by
// zipping b into a (pointers/lists a and b are changed)

    int counta = 0;
    int countb = 0;
    Node * tl = 0;
    // Node * tr = 0;

    Node * temp = a;
    while (temp->next != 0)
    {
        counta ++;
        temp = temp->next;
    }

temp = b;
while (temp->next != 0)
{
    countb ++;
    temp = temp->next;
}

for (int k = countb; k > 0; k--)
{
    tl = a;
    for (int a = 0; a < k; a++)
    {
        tl = tl->next;
        for (int b = 0; b < k; b++)
        {
            tl = tl->next;
            a = a + 1;
        }
    }
    tl = tl->next;
    a = tl;
}
return a;

Node * ziplist(Node * a, Node * b)
   // pre: 0 < size(a) and 0 < size(b) and size(b) < size(a)
   // post: returns pointer to first node of list created by
   // zipping b into a (pointers/lists a and b are changed)
{
    Node * p, * qa=a, * qb=b, * pb=a, * pq=a;
    while (pb != 0)
    {
        qa = qa->next;
        pb = pb->next;
        qa = qa->next;
        pb = pb->next;
        qa = qa->next;
        pb = pb->next;
    }
    return a;
}

http://www.cs.duke.edu/courses/cps100/current/inclass/link2/
Node * ziplist (Node * a, Node * b) {
    // pre: 0 < size(a) and 0 < size(b) and size(b) < size(a)
    // post: returns pointer to first node of list created by zipping b into a
    // (pointers/lists a and b are changed)

    Node * tempnext = 0;
    Node * tempb = b;
    Node * tempa = a;

    while (tempb != 0) {
        tempnext = tempa->next;
        tempa->next = tempb;
        tempb = tempb->next;
        tempa->next->next = tempnext;
        tempa = tempnext;
    }

    return a;
}