The Breakfast House

You are the owner of a restaurant "The Breakfast House" and want to calculate the sales in the restaurant. Your restaurant sells two dishes: Dosa[Rs.50] and Idli[Rs.40], represented by 0 and 1 respectively.

There is a queue of customers, each with a preferred dish to order.The kitchen in turn has a stack of prepared dishes ready to be served.As a customer moves to the start of the queue, he/she buys the dish on top of the kitchen stack if it matches his/her preference.If the dish doesn't match, he/she moves to the end of the queue without buying it.The process continues until either all customers get their preferred dish, or no one in the queue wants the dish on top of the kitchen stack.

Write a C program to calculate the total amount of money earned by the restaurant. If no customer wants the dish on top of the stack at any point, the program should return the money earned so far.

Input Format

1st line has the number of people(which is same as number of dishes in the stack)(n)

Next n lines have the dishes that each person wants to order

Next n lines have the dishes prepared in the kitchen.

Constraints

n >= 1

Elements in the stack and queue can be only 0 or 1

Output Format

Money earned by the restaurants as an integer

Sample Input 0

```
3
1
0
0
1
0
0
```

Sample Output 0

```
140
```

Explanation 0

Queue: 1 0 0

Stack: 0

0

1

(Element added in the last will be on the top)

The 1st person in the queue orders 1 but the stack has 0 so the person goes and stands behind the queue.

The queue now becomes 0 0 1

The person orders 0 and stack has 0 so he takes the order and pays 50 for the same.

Queue: 0 1

Stack: 0

1

The person wants to order 0 and 0 is on top of the stack, so he takes the order and pays 50.

Total = 100

Queue: 1

Stack:

1

The person wants 1 and has 1 on top of the stack, so he takes the order and pays 40.

Total money earned = 140

Sample Input 1

```
4
1
1
0
0
1
1
1
1
```

Sample Output 1

```
80
```

Explanation 1

Queue: 1 1 0 0

Stack:

1

1

1

1

The 1st person takes the order and pays 40.

Queue: 1 0 0

Stack:

1

1

1

The 2nd person in the queue takes the order and pays 40.

Total = 80

Queue: 0 0

Stack: 1

1

Both people in the queue don't want the item on top of the stack hence they don't take the order.

Total money earned = 80

C

```c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

void enqueue(int* queue, int q, int *f, int *r);
int dequeue(int* queue, int *f, int *r);
void push(int* stack, int q, int *top, int n);
void pop(int* stack, int *top, int *sum);
void costcalc(int *queue, int* stack, int *top, int *f, int *r, int *sum, int n);

int main()
{
    int n, i, f = 0, r = 0, top = -1, sum = 0;
    scanf("%d", &n); // Number of customers and dishes

    // Create queue for customer preferences and stack for prepared dishes
    int queue[n], stack[n];

    // Read customer preferences into queue
    for (i = 0; i < n; i++) {
        int pref;
        scanf("%d", &pref);
        enqueue(queue, pref, &f, &r);
    }

    // Read prepared dishes into stack
    for (i = 0; i < n; i++) {
        int dish;
        scanf("%d", &dish);
        push(stack, dish, &top, n);
    }

    // Calculate the total sales
    costcalc(queue, stack, &top, &f, &r, &sum, n);

    printf("%d\n", sum);
    return 0;
}

void enqueue(int* queue, int q, int *f, int *r) {
    queue[*r] = q;
    (*r)++;
}

int dequeue(int* queue, int *f, int *r) {
    int val = queue[*f];
    (*f)++;
    return val;
}
```

```c

void push(int* stack, int q, int *top, int n) {
    if (*top < n - 1) {
        (*top)++;
        stack[*top] = q;
    }
}

void pop(int* stack, int *top, int *sum) {
    if (*top >= 0) {
        if (stack[*top] == 0) {
            *sum += 50; // Dosa costs Rs. 50
        } else if (stack[*top] == 1) {
            *sum += 40; // Idli costs Rs. 40
        }
        (*top)--;
    }
}

void costcalc(int *queue, int* stack, int *top, int *f, int *r, int *sum, int n) {
    int mismatchCount = 0;

    while (*top >= 0 && mismatchCount < *r - *f) {
        if (queue[*f] == stack[*top]) {
            // If the customer's preference matches the top of the stack, pop and sell
            dequeue(queue, f, r);
            pop(stack, top, sum);
            mismatchCount = 0; // Reset mismatch counter
        } else {
            // If not, move the customer to the end of the queue
            enqueue(queue, dequeue(queue, f, r), f, r);
            mismatchCount++;
        }
    }
}
```

Line: 84 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔ Testcase 1 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
3
1
0
0
1
0
0
```

Your Output (stdout)

```
140
```

Expected Output

```
140
```