Practice problems for week 4

Question 1 - “Divisibility by 11”

Problem Description

Given an integer which has n digits, print `YES` if it is divisible by 11 and `NO` otherwise.

Note that the input number may contain leading zeroes.

Note that a number is divisible by 11 if and only if the difference of the sum of digits at odd positions and sum of digits at even positions in a number is divisible by 11.

Link to problem on OJ

Input Format

The first line of input contains a single integer n denoting the number of digits.

The second line of input contains a positive integer which is n digit long.

Input constraints

- $1 \leq n \leq 10000$

Output Format

Output `YES` if the number is divisible by 11 and `NO` otherwise.

Sample inputs and outputs

Sample Input 1

```
4
2548
```

Sample Output 1

```
NO
```

Sample Input 2

```
2
22
```

Sample Output 2

```
YES
```

Solution

```c
#include <stdio.h>

int main() {
    int n; scanf("%d\n", &n);
    int cnt[] = {0, 0};
    for (int i = 0; i < n; i++) {
        char c; scanf("%c", &c);
        cnt[i % 2] += (c - '0');
    }
    cnt[0] %= 11, cnt[1] %= 11;
    if ((cnt[0] - cnt[1] + 11) % 11) printf("NO\n");
    else printf("YES\n");
    return 0;
}
```

Question 2 - “Reversed” (Modified)

Problem Description

Given an integer input, print the number obtained by reversing its digits. If the reverse contains leading zeroes, **do not** output them

Input Format

The first and only line of input contains a positive integer N .

Input constraints

- $1 \leq N \leq 10^{18}$

Output Format

Output a single integer which denotes the integer obtained by reversing the digits of N , without any leading zeroes.

Sample inputs and outputs

Sample Input 1

```
348
```

Sample Output 1

```
843
```

Sample Input 2

```
23813900
```

Sample Output 2

```
931832
```

Solution

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

int main(void) {
    long long int n; scanf("%lld", &n);
    const int MAX_DIGITS = 20;
    int final[MAX_DIGITS];
    memset(final, 0, sizeof(final));
    int end = 0;
    while (n) {
        final[end++] = (n % 10);
        n /= 10;
    }
    int st = 0;
    while (!final[st]) st++; //would break if n = 0 was part of the input
    for (; st < end; st++) printf("%d", final[st]);
    printf("\n");
    return 0;
}
```