```c
#include <stdio.h>
int max(int a, int b) {
    return (a > b) ? a : b;
}
int longest_non_decreasing_subsequence(int seq[], int n) {
    int dp[n];
    int max_len = 1;
    for (int i = 0; i < n; i++) {
        dp[i] = 1;
    }
    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (seq[j] <= seq[i]) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
        if (dp[i] > max_len)
            max_len = dp[i];
    }
    return max_len;
}
int main() {
    int n;
    scanf("%d", &n);
    int seq[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &seq[i]);
    }
    printf("%d\n", longest_non_decreasing_subsequence(seq, n));
    return 0;
}
```

```c
#include <stdio.h>
#include <string.h>
int max(int a, int b) {
    return (a > b) ? a : b;
}
int longest_common_subsequence(char* s1, char* s2) {
    int len1 = strlen(s1);
    int len2 = strlen(s2);
    int dp[len1 + 1][len2 + 1];
    for (int i = 0; i <= len1; i++) {
        for (int j = 0; j <= len2; j++) {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (s1[i - 1] == s2[j - 1])
                dp[i][j] = dp[i - 1][j - 1] + 1;
            else
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
        }
    }
    return dp[len1][len2];
}
int main() {
    char s1[1001], s2[1001];
    scanf("%s %s", s1, s2);
    printf("%d\n", longest_common_subsequence(s1, s2));
    return 0;
}

```

	Input	Expected	Got	
✔	aab azb	2	2	✔

```c
#include <stdio.h>
int max(int a, int b) {
    return (a > b) ? a : b;
}
int main() {
    int n;
    scanf("%d", &n);
    int board[n][n];
    int dp[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &board[i][j]);
        }
    }
    dp[0][0] = board[0][0];
    for (int j = 1; j < n; j++) {
        dp[0][j] = dp[0][j-1] + board[0][j];
    }
    for (int i = 1; i < n; i++) {
        dp[i][0] = dp[i-1][0] + board[i][0];
    }
    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            dp[i][j] = board[i][j] + max(dp[i-1][j], dp[i][j-1]);
        }
    }
    printf("%d\n", dp[n-1][n-1]);
    return 0;
}
```

```c
#include <stdio.h>
unsigned long long count_ways(int n) {
    if (n < 0) return 0;
    unsigned long long dp[n + 1];
    for (int i = 0; i <= n; i++) dp[i] = 0;
    dp[0] = 1;
    for (int i = 1; i <= n; i++) {
        dp[i] += dp[i - 1];
        if (i >= 3) {
            dp[i] += dp[i - 3];
        }
    }
    return dp[n];
}
int main() {
    int n;
    scanf("%d", &n);
    if (n < 0) {
        printf("0\n");
        return 0;
    }
    printf("%llu\n", count_ways(n));
    return 0;
}

```

	Input	Expected	Got	
✔	6	6	6	✔
✔	25	8641	8641	✔
✔	100	24382819596721629	24382819596721629	✔