

```

1 #include <stdio.h>
2 int max(int a, int b) {
3     return (a > b) ? a : b;
4 }
5 int longest_non_decreasing_subsequence(int seq[], int n) {
6     int dp[n];
7     int max_len = 1;
8     for (int i = 0; i < n; i++) {
9         dp[i] = 1;
10    }
11    for (int i = 1; i < n; i++) {
12        for (int j = 0; j < i; j++) {
13            if (seq[j] <= seq[i]) {
14                dp[i] = max(dp[i], dp[j] + 1);
15            }
16        }
17        if (dp[i] > max_len)
18            max_len = dp[i];
19    }
20    return max_len;
21 }
22 int main() {
23     int n;
24     scanf("%d", &n);
25     int seq[n];
26     for (int i = 0; i < n; i++) {
27         scanf("%d", &seq[i]);
28     }
29     printf("%d\n", longest_non_decreasing_subsequence(seq, n));
30     return 0;
31 }

```

```

1 #include <stdio.h>
2 #include <string.h>
3 int max(int a, int b) {
4     return (a > b) ? a : b;
5 }
6 int longest_common_subsequence(char* s1, char* s2) {
7     int len1 = strlen(s1);
8     int len2 = strlen(s2);
9     int dp[len1 + 1][len2 + 1];
10    for (int i = 0; i <= len1; i++) {
11        for (int j = 0; j <= len2; j++) {
12            if (i == 0 || j == 0)
13                dp[i][j] = 0;
14            else if (s1[i - 1] == s2[j - 1])
15                dp[i][j] = dp[i - 1][j - 1] + 1;
16            else
17                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
18        }
19    }
20    return dp[len1][len2];
21 }
22 int main() {
23     char s1[1001], s2[1001];
24     scanf("%s %s", s1, s2);
25     printf("%d\n", longest_common_subsequence(s1, s2));
26     return 0;
27 }
28

```

	Input	Expected	Got	
✓	aab	2	2	✓
	azb			

```

1 #include <stdio.h>
2 int max(int a, int b) {
3     return (a > b) ? a : b;
4 }
5 int main() {
6     int n;
7     scanf("%d", &n);
8     int board[n][n];
9     int dp[n][n];
10    for (int i = 0; i < n; i++) {
11        for (int j = 0; j < n; j++) {
12            scanf("%d", &board[i][j]);
13        }
14    }
15    dp[0][0] = board[0][0];
16    for (int j = 1; j < n; j++) {
17        dp[0][j] = dp[0][j-1] + board[0][j];
18    }
19    for (int i = 1; i < n; i++) {
20        dp[i][0] = dp[i-1][0] + board[i][0];
21    }
22    for (int i = 1; i < n; i++) {
23        for (int j = 1; j < n; j++) {
24            dp[i][j] = board[i][j] + max(dp[i-1][j], dp[i][j-1]);
25        }
26    }
27    printf("%d\n", dp[n-1][n-1]);
28    return 0;
29 }

```

```

1 #include <stdio.h>
2 unsigned long long count_ways(int n) {
3     if (n < 0) return 0;
4     unsigned long long dp[n + 1];
5     for (int i = 0; i <= n; i++) dp[i] = 0;
6     dp[0] = 1;
7     for (int i = 1; i <= n; i++) {
8         dp[i] += dp[i - 1];
9         if (i >= 3) {
10             dp[i] += dp[i - 3];
11         }
12     }
13     return dp[n];
14 }
15 int main() {
16     int n;
17     scanf("%d", &n);
18     if (n < 0) {
19         printf("0\n");
20         return 0;
21     }
22     printf("%llu\n", count_ways(n));
23     return 0;
24 }
25

```

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