[**Dynamic Programming**](http://118.185.187.137/moodle/course/view.php?id=155#section-5)

**NAME:B Elumalai**

**ROLL No:230701084**

**1-DP-Playing with Numbers**



**Code**

#include <stdio.h>

long long count(int n) {

long long a[n + 1];

a[0] = 1;

for (int i = 1; i <= n; i++) {

a[i]=0;

a[i] += a[i - 1];

if (i >= 3) {

a[i] += a[i - 3];

}

}

return a[n];

}

int main() {

int n;

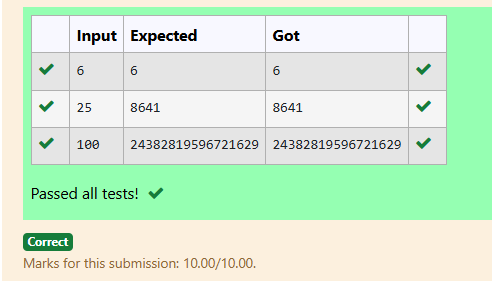
scanf("%d", &n);

printf("%lld\n", count(n));

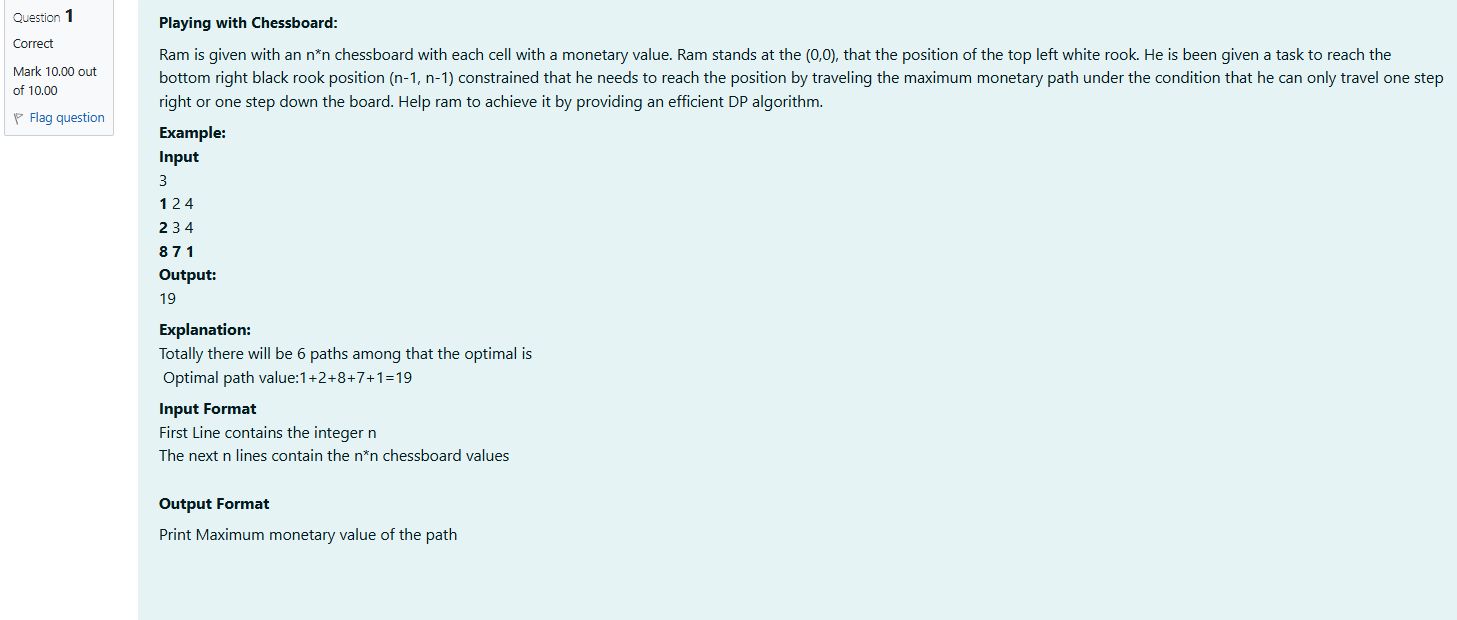
return 0;

}

**OUTPUT**

****

## 2-DP-Playing with chessboard

****

**CODE**

#include <stdio.h>

#define MAX 100

int main() {

int n, board[MAX][MAX], dp[MAX][MAX];

scanf("%d", &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] = (dp[i - 1][j] > dp[i][j - 1] ? dp[i - 1][j] : dp[i][j - 1]) + board[i][j];

}

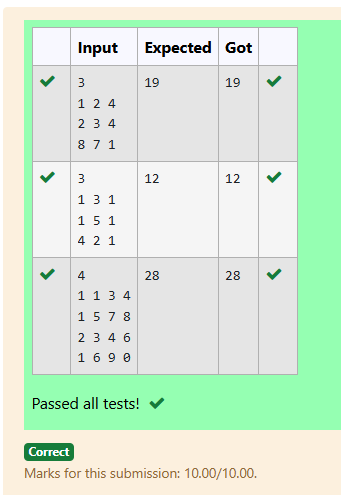
}

printf("%d\n", dp[n - 1][n - 1]);

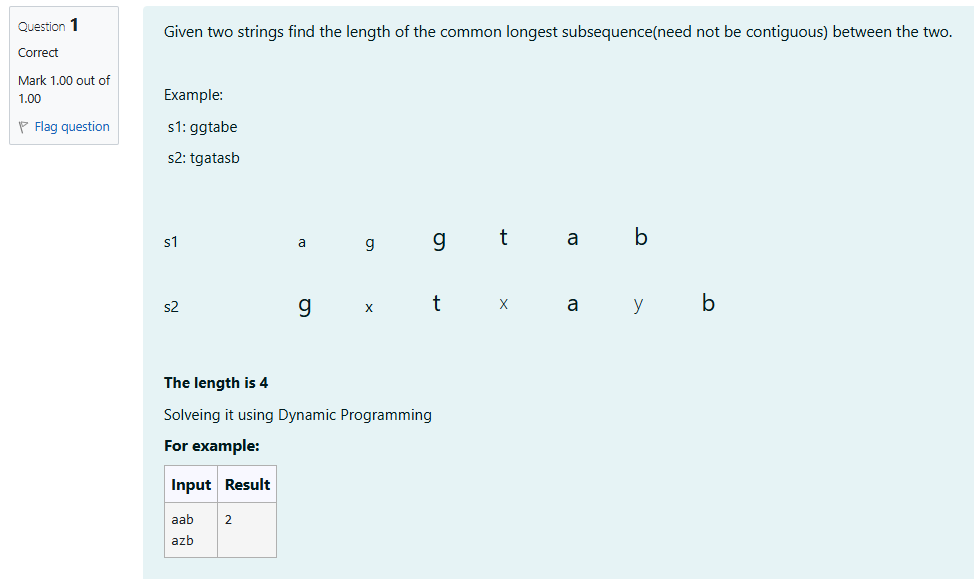
return 0;

}

**OUTPUT**

****

## 3-DP-Longest Common Subsequence

****

**CODE**

#include <stdio.h>

#include <string.h>

#define MAX\_LEN 100

int main() {

char s1[MAX\_LEN], s2[MAX\_LEN];

int dp[MAX\_LEN][MAX\_LEN];

scanf("%s", s1);

scanf("%s", s2);

int len1 = strlen(s1);

int len2 = strlen(s2);

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] = dp[i - 1][j] > dp[i][j - 1] ? dp[i - 1][j] : dp[i][j - 1];

}

}

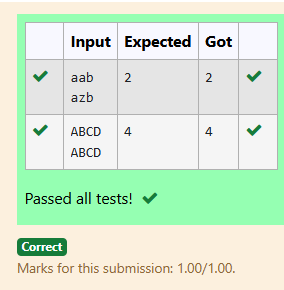
}

printf("%d\n", dp[len1][len2]);

return 0;

}

**OUTPUT**

****

## 4-DP-Longest non-decreasing Subsequence

## C:\Users\DELL\Pictures\Screenshots\Screenshot 2024-11-05 134852.png

## CODE

## #include <stdio.h>

## int longest\_non\_decreasing\_subsequence(int sequence[], int n) {

## int dp[n], max\_length = 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 (sequence[j] <= sequence[i]) {

## dp[i] = (dp[i] > dp[j] + 1) ? dp[i] : (dp[j] + 1);

## }

## }

## if (dp[i] > max\_length) {

## max\_length = dp[i];

## }

## }

## return max\_length;

## }

## int main() {

## int n;

## scanf("%d", &n);

## int sequence[n];

## for (int i = 0; i < n; i++) {

## scanf("%d", &sequence[i]);

## }

## int length = longest\_non\_decreasing\_subsequence(sequence, n);

## printf("%d\n", length);

## return 0;

## }

## OUTPUT

## C:\Users\DELL\Pictures\Screenshots\Screenshot 2024-11-05 135019.png