**DESIGN AND ANALYSIS OF ALGORITHM**

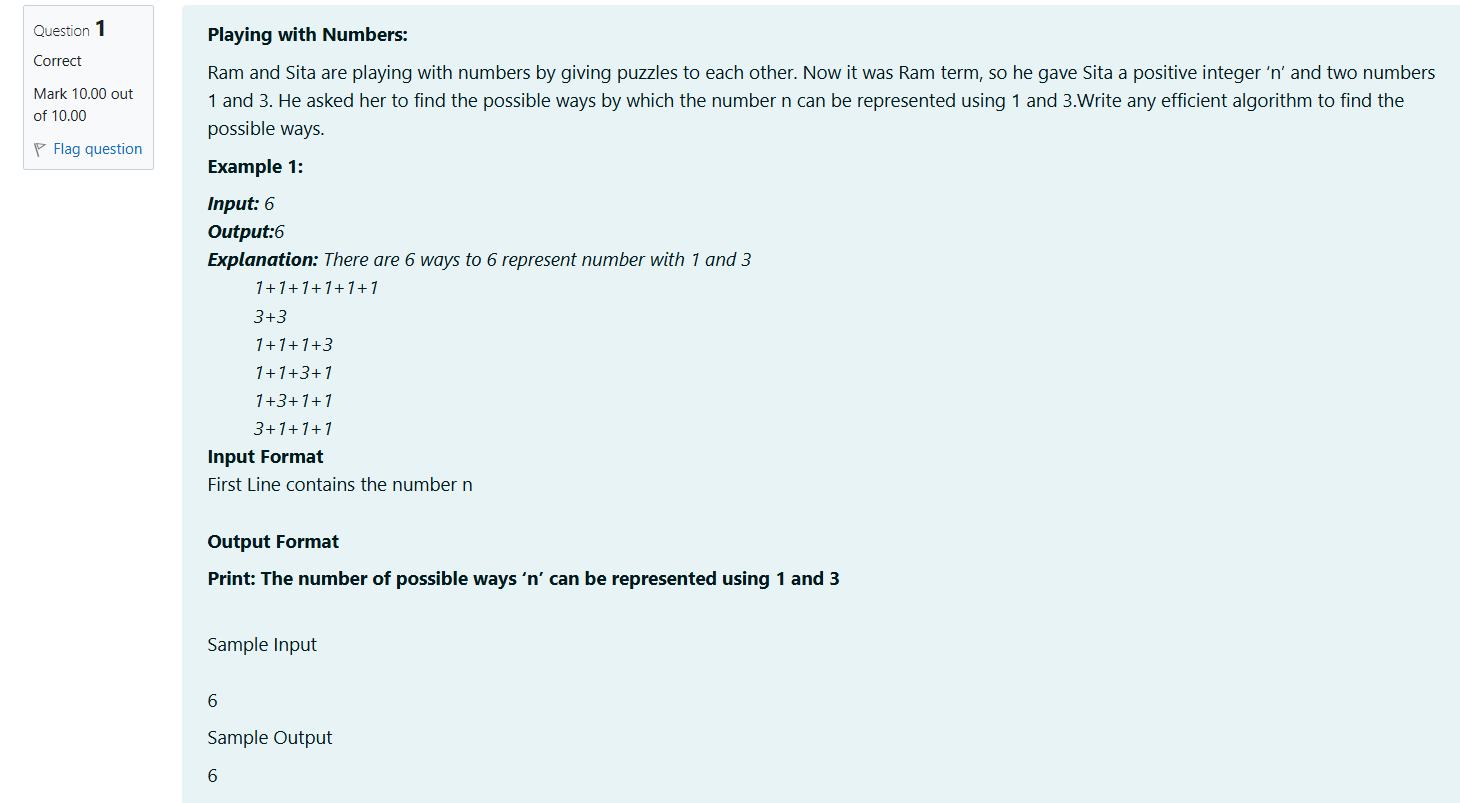
**DYNAMIC PROGRAMMING**

**NAME:ENIYA.B.A**

**CLASS: CSE-B**

**ROLL NO: 230701085**

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;

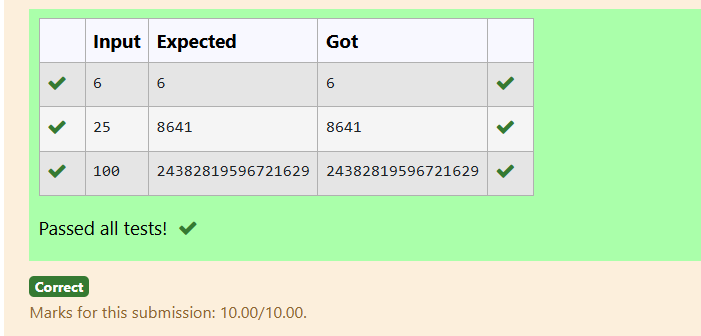
scanf("%d",&n);

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

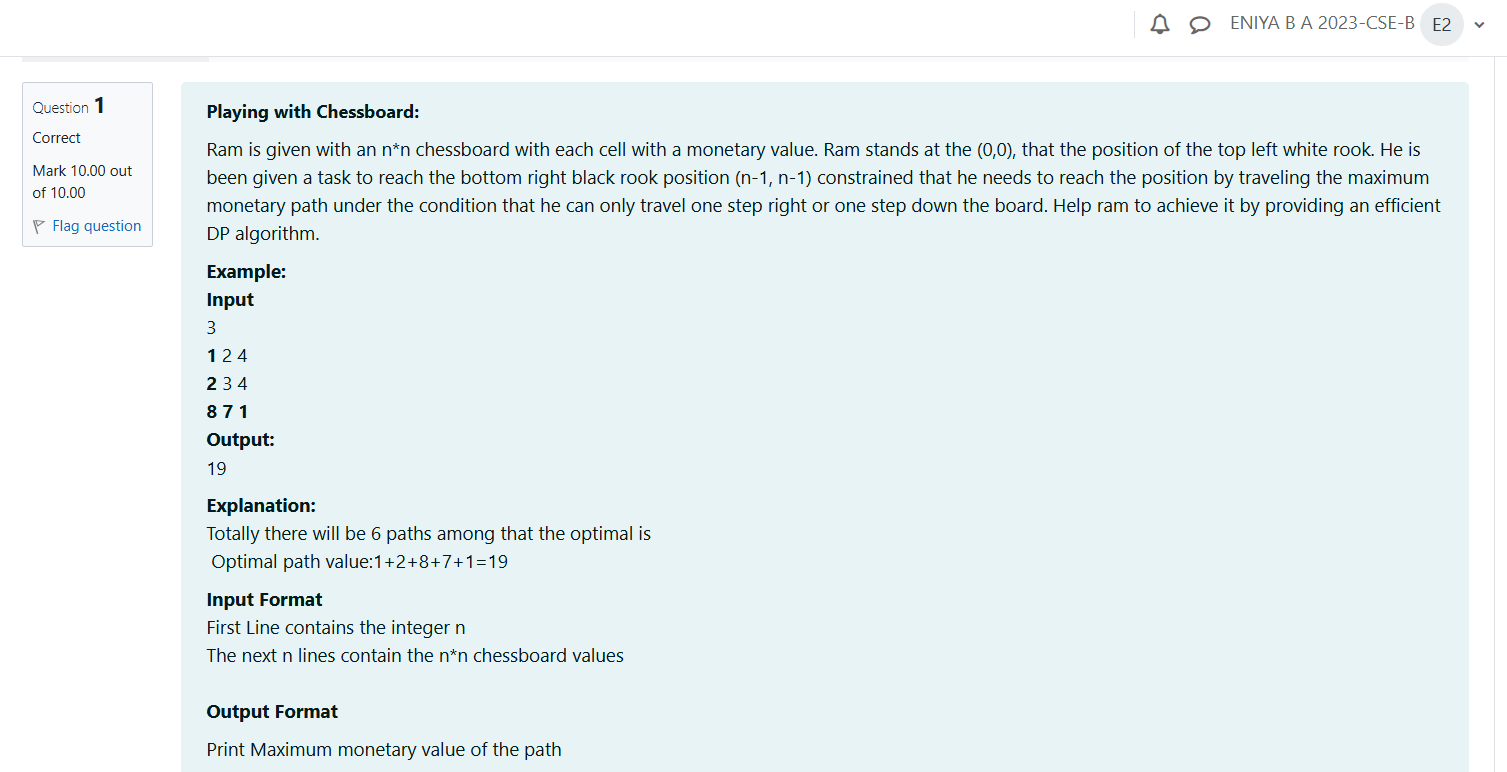
return 0;

}

OUTPUT:



## 2-DP-Playing with chessboard



CODE:

#include <stdio.h>

#define MAX 100

int main() {

int n;

int board[MAX][MAX];

int 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] = board[i][j] + (dp[i-1][j] > dp[i][j-1] ? dp[i-1][j] : dp[i][j-1]);}}

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

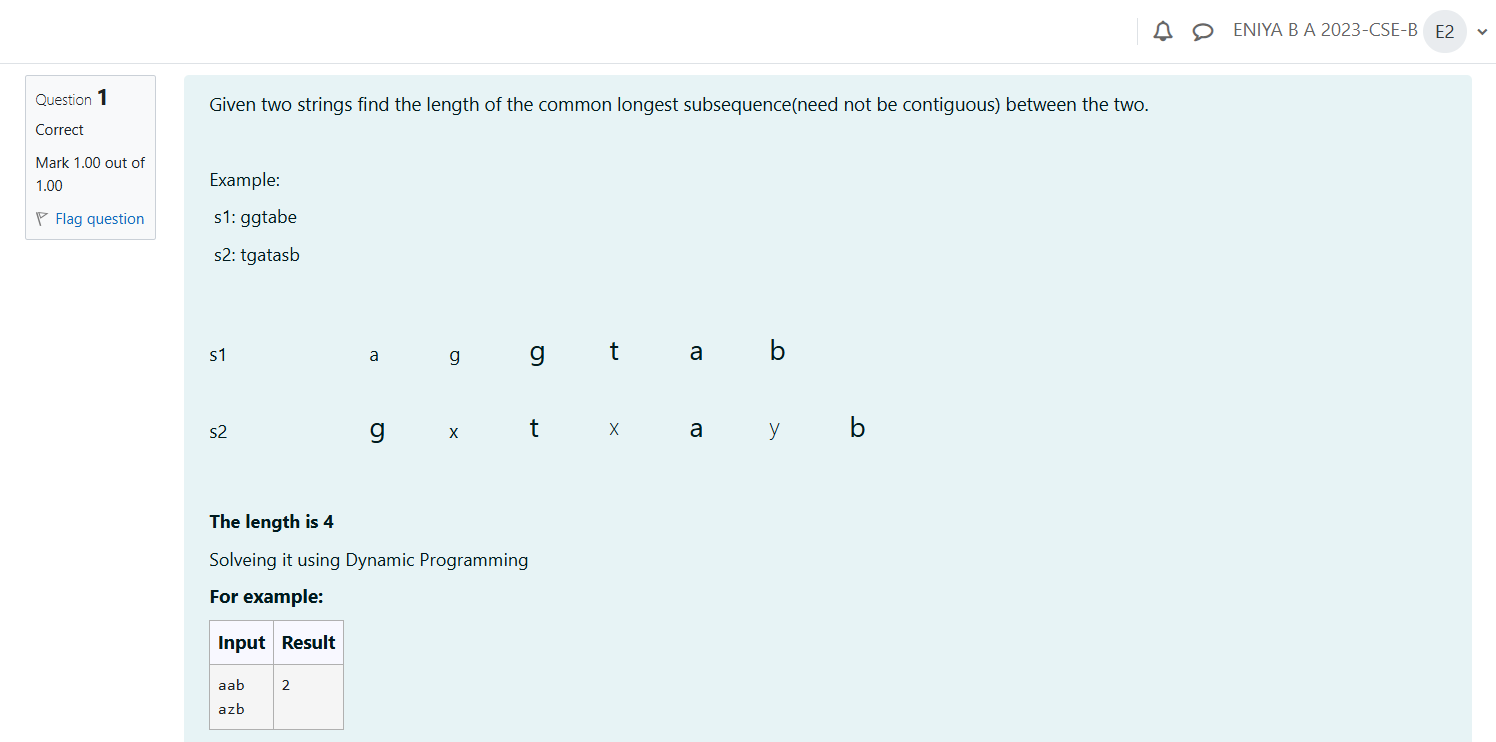
return 0;

}

OUTPUT:



3-DP-Longest Common Subsequence



CODE:

#include <stdio.h>

#include <string.h>

#define MAX 100

int main() {

char s1[MAX], s2[MAX];

int a[MAX][MAX];

scanf("%s", s1);

scanf("%s", s2);

int l1 = strlen(s1);

int l2 = strlen(s2);

for (int i = 0; i <= l1; i++) {

for (int j = 0; j <= l2; j++) {

if (i == 0 || j == 0) {

a[i][j] = 0;

} else if (s1[i - 1] == s2[j - 1]) {

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

} else {

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

}

}

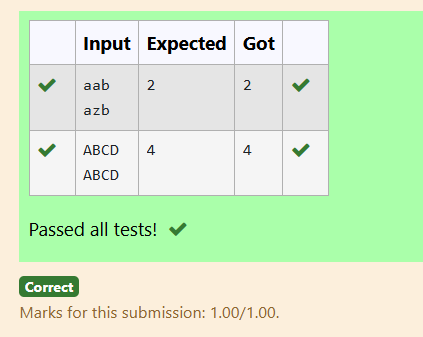
}

printf("%d\n",a[l1][l2]);

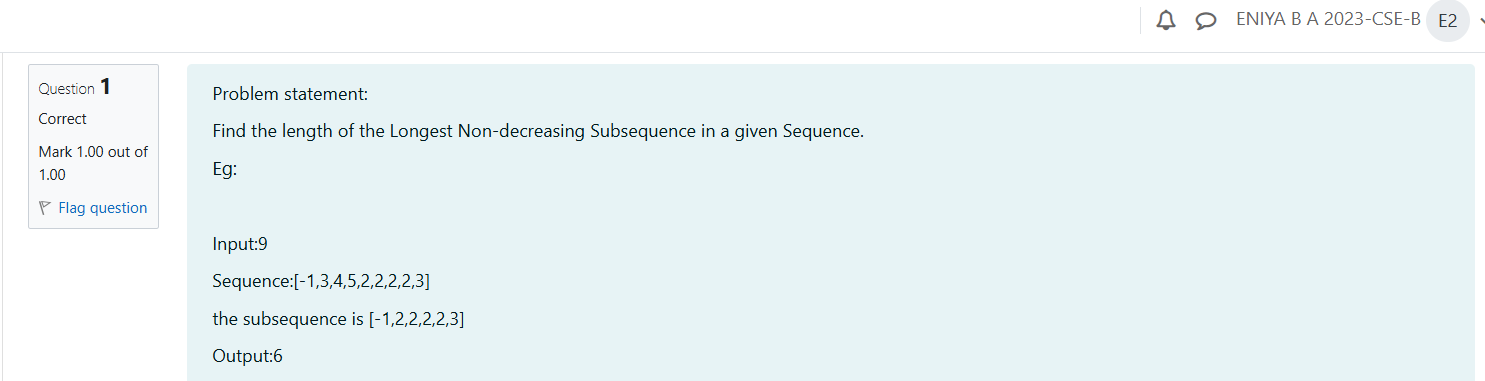
return 0;

}

OUTPUT:



## 4-DP-Longest non-decreasing Subsequence



CODE:

#include <stdio.h>

#define MAX 100

int main() {

int n;

int s[MAX];

int a[MAX];

scanf("%d",&n);

for (int i=0;i<n;i++) {scanf("%d",&s[i]);}

int m= 1;

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

a[i] = 1;

for (int j = 0; j < i; j++) {

if (s[j] <= s[i]) {a[i] = a[i]>a[j]+1?a[i]:a[j]+1;}

}

if (a[i] > m) {m = a[i];}

}

printf("%d\n", m);

return 0;

}

OUTPUT:

