Problem Result

Ways To Make Coin Change

Send Feedback

You are given an infinite supply of coins of each of denominations D = {D0, D1, D2, D3, ... Dn-1}. You need to figure out the total number of ways W, in which you can make change for Value V using coins of denominations D.

Note : Return 0, if change isn't possible.

Input Format

Line 1 : Integer n i.e. total number of denominations
Line 2 : N integers i.e. n denomination values
Line 3 : Value V

Output Format

Line 1 : Number of ways i.e. W

Constraints :

1<=n<=10
1<=V<=1000


```
// int help(int denominations[], int numDenominations, int value,int si,int**dp){

//     if(si == numDenominations+1)
//         return 0;
//     if(value ==0)
//         return 1;
//      if(dp[si][value]!=-1)
//         return dp[si][value];

//     int count = 0;
//     for(int i=0;i<=value;i++){
//         int temp = denominations[si]*i;
//         if(temp>value)
//             break;
//         count += help(denominations,numDenominations,value-temp,si+1,dp);
//     }
//     dp[si][value] =count;
//     return count;
// }



int countWaysToMakeChange(int denominations[], int numDenominations, int value){

  /*  Don't write main().
   *  Don't read input, it is passed as function argument.
   *  Return output and don't print it.
   *  Taking input and printing output is handled automatically.
   */
    int dp[value+1][numDenominations];

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

            if(i==0){
                dp[i][j] = 1;
                // continue;
            }else if(j==0){
                dp[i][j]=0;

            }else{

            dp[i][j]=0;
            for(int q=0;q<=j;q++){
                int temp = i*q;
                if(temp>j){
                    dp[i][j]=0;
                    break;
                }
                dp[i][j] += dp[i+1][temp];
            }
            }

        cout<<dp[i][j]<<" ";
        }
        cout<<endl;
    }
    return dp[0][value];



//   //numDenominations return help(denominations,numDenominations,value,0,output);

// }



int countWaysToMakeChange(int S[], int m, int n){
   *  Taking input and printing output is handled automatically.
   */

int i, j, x, y;

    // We need n+1 rows as the table is constructed
    // in bottom up manner using the base case 0
    // value case (n = 0)
    int table[n+1][m];

    // Fill the enteries for 0 value case (n = 0)
    for (i=0; i<m; i++)
        table[0][i] = 1;

    // Fill rest of the table entries in bottom
    // up manner
    for (i = 1; i < n+1; i++)
    {
        for (j = 0; j < m; j++)
        {
            // Count of solutions including S[j]
            x = (i-S[j] >= 0)? table[i - S[j]][j]: 0;

            // Count of solutions excluding S[j]
            y = (j >= 1)? table[i][j-1]: 0;

            // total count
            table[i][j] = x + y;
        }
    }
    return table[n][m-1];
}
```

CUSTOM INPUT SUBMIT SOLUTION