//C++ implementation for subset sum

// problem using recursion

#include <bits/stdc++.h>

using namespace std;

// Function to check if there is a subset

// with the given sum using recursion

bool isSubsetSumRec(vector<int>& arr, int n, int sum) {

// Base Cases

if (sum == 0)

return true;

if (n == 0)

return false;

// If last element is greater than sum,

// then ignore it

if (arr[n - 1] > sum)

return isSubsetSumRec(arr, n - 1, sum);

// Check if sum can be obtained by including

// or excluding the last element

return isSubsetSumRec(arr, n - 1, sum)

|| isSubsetSumRec(arr, n - 1, sum - arr[n - 1]);

}

bool isSubsetSum(vector<int>& arr, int sum) {

return isSubsetSumRec(arr, arr.size(), sum);

}

int main() {

vector<int> arr = {3, 34, 4, 12, 5, 2};

int sum = 9;

if (isSubsetSum(arr, sum))

cout << "True" << endl;

else

cout << "False" << endl;

return 0;

}