这个问题又称为“子集和问题”（也就是给定一个整数集合和一个定值，从一个集合中选取一个子集，使得子集中所有数的和等于给定的值，具体的可以百度，google 子集和问题），这是一个NP完全问题，不存在多项式时间的解，所以没有好的算法。

回朔法（本质还是穷举法）

动态规划（两种）  
回朔法（第一种）  
#include <iostream>

#include <vector>

#include <bitset>

**using** **namespace** std;

void print(bitset<5> & X)

{

**for** (int i = 0; i < X.size(); ++i)

{

cout << X[i] << " ";

}

cout << endl;

}

void subsetSum(vector<int>& W, bitset<5>& X, int sum, int targetSum, int k)

{

X[k] = 1; // try one branch of tree

**if** (sum + W[k] == targetSum)

print(X); // we have a solution

**else** **if** (k + 1 < W.size() && sum + W[k] <= targetSum) // include the W[k]

subsetSum(W, X, sum + W[k], targetSum, k + 1); // the maxium k is W.size-2

**if** (k + 1 < W.size() && sum + W[k + 1] <= targetSum) {

X[k] = 0; // uniclude the W[k]

subsetSum(W, X, sum, targetSum, k + 1); // the maxium k is W.size-2

}

}

int main() {

vector<int> W = { 2, 4, 6, 8, 10 };

bitset<5> X;

subsetSum(W, X, 0, 20, 0);

system("pause");

**return** 0;

}