#include<bits/stdc++.h>

using namespace std;

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

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

for(int j=1;j<=sum;j++){

if(arr[i-1]<=j){

dp[i][j] = dp[i-1][j-arr[i-1]] || dp[i-1][j];

}

else{

dp[i][j] = dp[i-1][j];

}

}

}

return dp[n][sum];

}

int main(){

int n,sum=0;

cin>>n;

vector<int> arr(n);

int i = n;

int j = 0;

while(i--){

int x;

cin>>x;

arr[j] = x;

sum+=x;

j++;

}

int range = sum/2;

vector<vector<bool> > dp(n+1,vector<bool> (range+1,-1));

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

dp[0][i] = false;

}

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

dp[i][0] = true;

}

bool a = subset(arr,range,dp,n);

vector<int> v;

int mm = INT\_MIN;

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

if(dp[n][i]){

// v.push\_back(i);

mm = max(mm,i);

}

}

int ans = sum - (2\*mm);

cout<<ans;

}

------------------------------------------------------------

#include<bits/stdc++.h>

using namespace std;

bool subset(vector<int> arr, int sum, vector<vector<bool> > dp, int n){

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

for(int j=1;j<=sum;j++){

if(arr[i-1]<=j){

dp[i][j] = dp[i-1][j-arr[i-1]] || dp[i-1][j];

}

else{

dp[i][j] = dp[i-1][j];

}

}

}

return dp[n][sum];

}

int main(){

int n,sum=0;

cin>>n;

vector<int> arr(n);

int i = n;

int j = 0;

while(i--){

int x;

cin>>x;

arr[j] = x;

sum+=x;

j++;

}

// vector<vector<bool> > dp(n+1,vector<bool> (sum+1,-1));

// for(int i=0;i<=sum;i++){

// dp[0][i] = false;

// }

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

// dp[i][0] = true;

// }

int max\_x = INT\_MIN;

for(int irange=0;irange<=sum/2;irange++){

vector<vector<bool> > dp(n+1,vector<bool> (irange+1,-1));

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

dp[0][i] = false;

}

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

dp[i][0] = true;

}

if(subset(arr,irange,dp,n)){

max\_x = max(max\_x,irange);

}

}

int ans = sum - (2\*max\_x);

cout<<ans;

}