#include<bits/stdc++.h>

using namespace std;

// **remember to make dp array in int in recursion to check whether dp[n][sum] is updated or not**

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

if(dp[n][sum]==-1){

if(arr[n-1]<=sum){

dp[n][sum] = max(subset(arr,sum-arr[n-1],dp,n-1),subset(arr,sum,dp,n-1));

}

else{

dp[n][sum] = subset(arr,sum,dp,n-1);

}

}

return dp[n][sum];

}

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

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

dp[0][i] = false;

}

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

dp[i][0] = true;

}

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;

cin>>n>>sum;

vector<int> arr(n);

int i = n;

int j = 0;

while(i--){

int x;

cin>>x;

arr[j] = x;

j++;

}

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

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

cout<<"TRUE"<<endl;

}

else{

cout<<"FALSE"<<endl;

}

}