题目大意：一张CD上有不超过20首歌，选取时间和不超过n分钟的歌曲，要求时间和最大，最后按原顺序输出每首歌曲的时间，及时间和。

根据01背包二维数组的动态转移方程dp[i][j]=max(dp[i-1][j],dp[i-1][j-v[i]]+w[i]),可以知道，dp[i][j]的状态和dp[i-1][j]、dp[i-1][j-v[i]]有关，于是，在记录路径的时候，要是dp[i][j]==dp[i-1][j]，说明，这条路没有走向dp[i-1][j-v[i]]，这时，可以另开一个数组a[i][j]，当dp[i][j]在动态转移时，==dp[i-1][j],则a[i][j]=0,否则a[i][j]=1;然后对数组a[i][j]进行回溯即可，具体看代码

#include<iostream>

#include<cstdio>

#include<algorithm>

#include<cstring>

#include<iomanip>

#include<cstring>

#include<cstdlib>

#include<cstring>

#define ll long long

using namespace std;

int num;

bool a[30][10010];

int dp[10010],w[10010],p[10010];

int main()

{

//freopen("input.txt","r",stdin);

int m,n;

while(cin>>m>>n)

{

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

cin>>w[i];

memset(dp,0,sizeof(dp));

memset(a,0,sizeof(a));

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

for(int j=m;j>=w[i];j--)

if(dp[j-w[i]]+w[i]>=dp[j])

{

dp[j]=dp[j-w[i]]+w[i];

a[i][j]=1;

}

for(int i=n-1,j=m;i>=0;i--)

if(a[i][j]==1)

{

cout<<w[i]<<' ';

j-=w[i];

}

cout<<"sum:"<<dp[m]<<"\n";

}

return 0;

}