#include<bits/stdc++.h>

using namespace std;

int knapsack(vector<int> wt, vector<int> cost, int W, vector<vector<int> > &dp, int n){

if(n==0 || W==0){

dp[n][W] = 0;

}

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

if(wt[n-1]<=W){

dp[n][W] = max(cost[n-1]+knapsack(wt,cost,W-wt[n-1],dp,n-1), knapsack(wt,cost,W,dp,n-1));

}

else{

dp[n][W] = knapsack(wt,cost,W,dp,n-1);

}

}

return dp[n][W];

}

int main(){

int n,W;

cin>>n>>W;

vector<int> cost(n);

vector<int> wt(n);

int i = n;

int j = 0;

while(i--){

int x,y;

cin>>x>>y;

wt[j] = x;

cost[j] = y;

j++;

}

vector<vector<int> > dp(n+1,vector<int> (W+1,-1));

cout<<knapsack(wt,cost,W,dp,n);

}