#include <iostream>

using namespace std;

int unboundedknapSack(int w,int wt[],int val[],int n){

int dp[n+1][w+1];

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

for(int j=0;j<=w;j++){

if(i==0 || j==0){

dp[i][j]=0;

}

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

dp[i][j]=max(val[i-1]+dp[i][j-wt[i-1]], dp[i-1][j]);

}

else{

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

}

}

}

return dp[n][w];

}

int main() {

int val[] = {10,30,20};

int wt[] = {5, 10, 15};

int w = 100;

int n = sizeof(val)/sizeof(val[0]);

cout<<unboundedknapSack(w, wt, val, n);

return 0;

}