0/1 KNAPSACK

Question :

There are N items, numbered 1, 2, ... N. For each i $(1 \le i \le N)$, Item i has a weight of W_i and a value of V_i. Taro has decided to choose ~~one~~ same of the N items and carry them home in a knapsack. The capacity of the Knapsack is W, which means that the sum of the weights of items taken must be at most W.

Find the maximum possible sum of the values of items that Taro takes home.

Input format:

```
N   W
W1  V1
W2  V2
|   |
|   |
WN  VN
```

CODE

```
INT WT[105], VAL[105];
LL DP[105][100005];

LL FUNC(INT IND, INT WT_LEFT)
{
    -> Base cases
    IF (WT_LEFT == 0)  RETURN 0;
    IF (IND < 0)  RETURN 0;
    IF (DP[IND][WT_LEFT] != -1)
        RETURN DP[IND][WT_LEFT];

    -> Don't choose item at ind index
    LL ANS = FUNC(IND -1, WT-LEFT);

    -> Choose item at ind index
    IF (WT_LEFT -WT[IND] >= 0)
        ANS = MAX(ANS, FUNC(IND -1, WT_LEFT
            -WT[IND]) + VAL[IND]);
    RETURN DP[IND][WT-LEFT] = ANS;
}

INT MAIN()
{
    MEMSET(DP, -1, SIZEOF(DP));
    INT n, W;
    CIN >> n >> W;
    FOR (INT i=0; i< n; i++)
    {
        CIN >> WT[i] >> VAL[i];
    }
    COUT << FUNC(n-1, W) << "\n";

    RETURN 0;
}
```

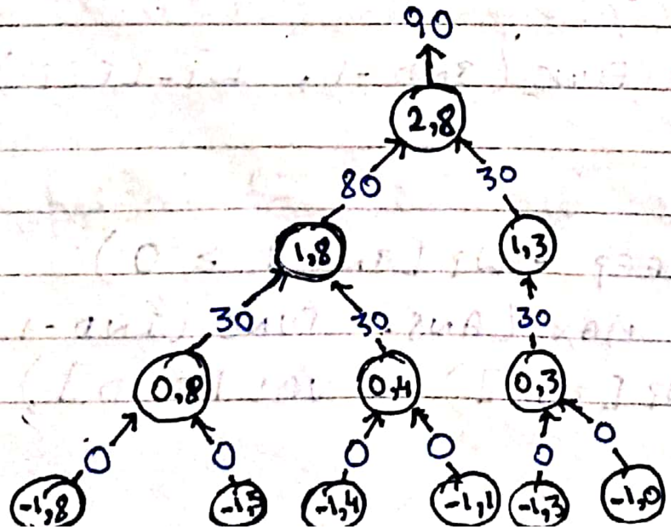
INPUT:

```
3 8
3 30
4 50
5 60
```

OUTPUT:

```
90
```

RECURSION TREE (For given input) - W/T DP



90
2,8
80
30
1,8
1,3
30
30
30
0,8
0,4
0,3
0
0
0
0
0
0
-1,8
-1,5
-1,4
-1,1
-1,3
-1,0