#include <iostream>

#include <opencv2/opencv.hpp>

using namespace cv;

using namespace std;

# define MAX 100

int front[MAX][MAX];

void trace(int m[MAX][MAX], int w[MAX], int c, int n, int x[MAX])

{

for (int i = 1; i < n; i++)

{

if (m[i][c] == m[i + 1][c])

{

x[i] = 0;

}

else

{

x[i] = 1;

c = c - w[i];

}

}

x[n] = (m[n][c]) ? 1 : 0;

}

void Draw(int m[MAX][MAX], int front[MAX][MAX], int n, int c)

{

string temp[MAX][MAX];

for (int i = 1; i <= n; i++)

{

for (int j = 0; j <= c; j++)

{

char s[6];

sprintf\_s(s, "%d", m[i][j]);

temp[i][j] = s;

}

}

for (int i = n; i >= 1; i--)

{

for (int j = 0; j <= c; j++)

{

Point p(j \* 80, 100 \* i);

putText(src, temp[i][j], p, FONT\_HERSHEY\_SIMPLEX, 1, Scalar(0, 0, 0), 4, 8);

}

imshow("input image", src);

waitKey(1000);

if (i < n)

{

for (int j = 0; j <= c; j++)

{

Point p1(front[i][j] \* 80 + 10, 100 \* (i + 1) - 30);

Point p2(80 \* j + 10, 100 \* i + 10);

arrowedLine(src, p1, p2, Scalar(28, 28, 236), 2, LINE\_AA, 0, 0.05);

}

}

imshow("input image", src);

waitKey(1000);

}

waitKey(0);

}

void knain(int n, int c, int v[MAX], int w[MAX], int m[MAX][MAX])

{

for (int i = n; i >= 1; i--)

{

int jmax = min(w[i] - 1, c);

for (int j = 0; j <= jmax; j++)

{

m[i][j] = m[i + 1][j];

front[i][j] = j;

}

for (int j = w[i]; j <= c; j++)

{

m[i][j] = max(m[i + 1][j], m[i + 1][j - w[i]] + v[i]);

if (m[i][j] == m[i + 1][j])

{

front[i][j] = j;

}

else

{

front[i][j] = j - w[i];

}

}

}

for (int i = 1; i <= n; i++)

{

for (int j = 0; j <= c; j++)

{

printf("%3d", m[i][j]);

}

printf("\n");

}

}

int main()

{

Mat src = imread("D:/Desktop/white.png");

int n, c;

scanf\_s("%d", &n);

scanf\_s("%d", &c);

int v[MAX] = { 0 }, w[MAX] = { 0 }, x[MAX] = { 0 };

for (int i = 1; i <= n; i++)

{

scanf\_s("%d", &w[i]);

scanf\_s("%d", &v[i]);

}

int m[MAX][MAX] = { 0 };

knain(n, c, v, w, m);

trace(m, w, c, n, x);

Draw(m, front, n, c);

printf("\n%d\n", m[1][c]);

for (int i = 1; i <= n; i++)

{

printf("%d ", x[i]);

}

return 0;

}

