

```

n := 3;
m := 4;
A := [20, 15, 25];
B := [13, 17, 19, 11];
C := [[6, 5, 2, 1], [3, 5, 4, 2], [5, 3, 6, 3]];
X := [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]];
#Метод на Северозападния ъгъл
i := 0;
j := 0;

```

докато е истина прави:

```

    ако A[i] = B[j] и i = n - 1 и j = m - 1 то:
        X[i][j] := A[i];

```

спри;

```

    ако A[i] > B[j] то:
        X[i][j] := B[j];
        A[i] := A[i] - B[j];
        j := j + 1;

```

```

    ако A[i] < B[j] то:
        X[i][j] := A[i];
        B[j] := B[j] - A[i];
        i := i + 1;

```

```

    ако A[i] = B[j] то:
        X[i][j] := A[i];
        i := i + 1;
        j := j + 1;

```

отпечатай(X, '\n');

s := 0;

докато е истина прави:

```

    Empty_Cells = [];

```

за i от 0 до n*m-(n+m-1) прави:

```

        Empty_Cells[i] := 0;

```

```

    Full_Cells = [];

```

за i от 0 до n + m - 1 прави:

```

        Full_Cells[i] := 0;

```

```

    p := 0;

```

```

    q := 0;

```

за i от 0 до n прави:

за j от 0 до m прави:

```

        ако X[i][j] = 0 то:
            Empty_Cells[p] := [i, j];
            p = p + 1;
        ако X[i][j] <> 0 то:
            Full_Cells[q] := [i, j];
            q = q + 1;

```

```

    Loop = [];

```

за k от 0 до размера на Empty_Cells прави:

```

        Loop[] := [];

```

за k от 0 до размера на Empty_Cells прави:

```

        i0 := Empty_Cells[k][0];
        j0 := Empty_Cells[k][1];
        Edge_k := [i0, j0];
        Loop[k] := Edge_k;

```

за r от 0 до размера на Full_Cells прави:

ако j0 = Full_Cells[r][1] мо:
Edge_k := Full_Cells[r];
Loop[k] := Edge_k;
i1 := Full_Cells[r][0];
спри;

за r от 0 до размера на Full_Cells прави:

ако i1 = Full_Cells[r][0] и j0 <> Full_Cells[r][1] мо:
Edge_k := Full_Cells[r];
Loop[k] := Edge_k;
j1 := Full_Cells[r][1];
спри;

за r от 0 до размера на Full_Cells прави:

ако [i0, j1] = Full_Cells[r] мо:
Edge_k := [i0, j1];
Loop[k] := Edge_k;
спри;

ако размера на Loop[k] = 2 мо:

изтриваме Loop[k][1];

ако размера на Loop[k] < 4 мо:

за r от 0 до размера на Full_Cells прави:
if j1 == Full_Cells[r][1] and i1 != Full_Cells[r][0]:
Edge_k = Full_Cells[r];
Loop[k].append(Edge_k);
i2 = Full_Cells[r][0];
break;

за r от 0 до размера на Full_Cells прави:

ако i2 = Full_Cells[r][0] и j1 <> Full_Cells[r][1] мо:
Edge_k := Full_Cells[r];
Loop[k] := Edge_k;
j2 = Full_Cells[r][1];
спри;

за r от 0 до размера на Full_Cells прави:

ако [i0, j2] = Full_Cells[r] мо:
Edge_k := [i0, j2];
Loop[k] := Edge_k;
спри;

ако k = len(Empty_Cells) - 1 и s = 0 мо:

s := s + 1;
Loop[k] := [];
i0 := Empty_Cells[k][0];
j0 := Empty_Cells[k][1];
Edge_k := [i0, j0];
Loop[k] := Edge_k;

за r от размера на Full_Cells до 0 прави:

ако j0 = Full_Cells[r][1] мо:
Edge_k := Full_Cells[r];
Loop[k] := Edge_k;
i1 := Full_Cells[r][0];
спри;

за r от размера на Full_Cells до 0 прави:

ако i1 = Full_Cells[r][0] и j0 <> Full_Cells[r][1] то:

Edge_k := Full_Cells[r];

Loop[k] := Edge_k;

j1 := Full_Cells[r][1];

спи;

за r от размера на Full_Cells до 0 прави:

ако [i0, j1] = Full_Cells[r] то:

Edge_k := [i0, j1];

Loop[k] := Edge_k;

спи;

Q := 10**5;

u := [];

за i от 0 до n прави:

u[i] := Q;

v := [];

за i от 0 до m прави:

v[i] := Q;

u[Full_Cells[0][0]] := 0;

за i от 0 до размера на Full_Cells прави:

ако u[Full_Cells[i][0]] <> Q и v[Full_Cells[i][1]] = Q то:

v[Full_Cells[i][1]] := C[Full_Cells[i][0]][Full_Cells[i][1]] - u[Full_Cells[i][0]];

ако v[Full_Cells[i][1]] <> Q и u[Full_Cells[i][0]] = Q то:

u[Full_Cells[i][0]] := C[Full_Cells[i][0]][Full_Cells[i][1]] - v[Full_Cells[i][1]];

за i от 0 до размера на Full_Cells прави:

ако u[Full_Cells[i][0]] <> Q и v[Full_Cells[i][1]] = Q то:

v[Full_Cells[i][1]] := C[Full_Cells[i][0]][Full_Cells[i][1]] - u[Full_Cells[i][0]];

ако v[Full_Cells[i][1]] <> Q и u[Full_Cells[i][0]] = Q то:

u[Full_Cells[i][0]] := C[Full_Cells[i][0]][Full_Cells[i][1]] - v[Full_Cells[i][1]];

Delta := [];

delta_k := 0;

за i от 0 до размера на Empty_Cells прави:

delta_k := u[Empty_Cells[i][0]] + v[Empty_Cells[i][1]] -
C[Empty_Cells[i][0]][Empty_Cells[i][0]];

Delta[i] := delta_k;

delta_k := 0;

max_elem := Delta[0];

за i от 0 до размера на Delta прави:

ако max_elem < Delta[i] то:

max_elem := Delta[i];

ако max_elem <= 0 то:

спи;

index_max_elem := индекса на max_elem от Delta;

min_x := [];

за j от 0 до размера на Loop[index_max_elem] прави:

k1 := Loop[index_max_elem][j][0];

k2 := Loop[index_max_elem][j][1];

ако j разделено с остатък на 2 <> 0 то:

min_x[j] := X[k1][k2];

за j от 0 до размера на Loop[index_max_elem] прави:

k1 := Loop[index_max_elem][j][0];

k2 := Loop[index_max_elem][j][1];

ако j разделено с остатък на 2 = 0 то:

X[k1][k2] := X[k1][k2] + min(min_x);

иначе:

X[k1][k2] := X[k1][k2] - min(min_x);

отпечатай(Delta, '\n');

отпечатай(X, '\n');

F := 0;

за i от 0 до n прави:

за j от 0 до t прави:

ако X[i][j] <> 0 то:

F := F + X[i][j]*C[i][j];

отпечатай(F);