In [1]:
```
import numpy as np
import pandas as pd
```

Инициализируем исходные данные по задаче (вариант 23).

In [2]:
```
m = 8
p = 12
f = pd.Series([12, 10, 8, 6, 4, 2, 0, 0, 0], name='f_from_t')
f
```

Out[2]:
```
0    12
1    10
2     8
3     6
4     4
5     2
6     0
7     0
8     0
Name: f_from_t, dtype: int64
```

In [53]:
```
frame = pd.concat({(m, 'B'): f.copy(), (m, 'x'): f.apply(lambda x: 0 if x > 0 else (0,5 if x == 0 else 1))}, ax
frame.index.name = 't'
frame.columns.names = ['i', 'step_vals']
frame
```

Out[53]:

i	8	
step_vals	B	x
t		
0	12	0
1	10	0
2	8	0
3	6	0
4	4	0
5	2	0
6	0	(0, 5)
7	0	(0, 5)
8	0	(0, 5)

In [54]:
```
def get_B_val(row: pd.Series):
    if row.name == m:
        next_B = 0
    else:
        next_B = frame.iloc[row.name + 1, -2]

    return max(f[row.name] + next_B, frame.iloc[1, -2])

def get_x_val(row: pd.Series):
    if row.name == m:
        next_B = 0
    else:
        next_B = frame.iloc[row.name + 1, -2]

    if f[row.name] + next_B > frame.iloc[1, -2]:
        return 0
    elif f[row.name] + next_B < frame.iloc[1, -2]:
        return 1
    else:
        return 0,5

for i in np.arange(m-1, 0, -1):
    step_B = []
    step_x = []
    for t in np.arange(m + 1):
        step_B.append(get_B_val(frame.loc[t]))
        step_x.append(get_x_val(frame.loc[t]))
    frame = frame.merge(pd.Series(step_B, name=(i, 'B')), left_index=True, right_index=True)
    frame = frame.merge(pd.Series(step_x, name=(i, 'x')), left_index=True, right_index=True)
frame
```

Out[54]:

i	8		7		6		5		4		3		2		1
step_vals	B	x	B	x	B	x	B	x	B	x	B	x	B	x	B	x
t																
0	12	0	22	0	30	0	36	0	40	0	48	0	54	0	60	0
1	10	0	18	0	24	0	28	0	36	0	42	0	48	0	54	0
2	8	0	14	0	18	(0, 5)	26	0	32	0	38	0	44	0	50	0
3	6	0	10	(0, 5)	18	1	24	(0, 5)	30	0	36	1	42	(0, 5)	48	(0, 5)
4	4	0	10	1	18	1	24	1	28	(0, 5)	36	1	42	1	48	1
5	2	0	10	1	18	1	24	1	28	1	36	1	42	1	48	1
6	0	(0, 5)	10	1	18	1	24	1	28	1	36	1	42	1	48	1
7	0	(0, 5)	10	1	18	1	24	1	28	1	36	1	42	1	48	1
8	0	(0, 5)	10	1	18	1	24	1	28	1	36	1	42	1	48	1