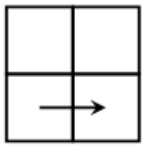
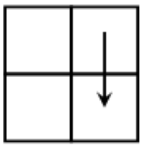
CSC 323 Algorithm Design and Analysis, Fall 2019
Instructor: Dr. Natarajan Meghanathan
Project 5: Dynamic Programming Algorithm for Optimum Coin Collection in a Two-Dimensional Grid

Due: Oct. 22nd, 11.59 PM (Submission through Canvas)

In this project, you will extend the dynamic programming algorithm that we discussed in class for the Coin Collection problem in a two-dimensional grid and implement the same.

The conditions for the robot movement are as follows: at any time, the robot can move one cell down or one cell to the right


One cell to the right


One cell down

Each of you are assigned a grid of dimensions n (rows) x m (columns) as specified in the next page. You are required to randomly distribute P number of coins (where P < n*m) across the cells of the grid (at most one coin per cell). The value for a coin assigned to a cell is randomly chosen from the range 1 ... V. The P and V values are also assigned specifically for each student.

Your tasks are as follows:
(1) Implement the dynamic programming algorithm to calculate the optimum (maximum) value of the coins that a robot could collect as it traverses from cell (0, 0) to any cell in the grid such that at any time, a robot can have one of the two movements mentioned above.
(2) Extend the dynamic programming algorithm to also keep track of the path traced by the robot to reach any target cell in the grid starting from cell (0, 0).
(3) As output, your code should print the following:

(i) The optimum value of the coins that a robot could collect to reach any target cell in the grid starting from cell (0, 0), as shown in the table sample output (see next page).
(ii) The sequence of cells that the robot should visit to collect the optimum value of the coins starting from cell (0, 0) to cell (n-1, m-1).

Submission (in Canvas):
1) The entire code as a .cpp file.
2) Include a screenshot (as shown in a sample output displayed in the next page) of the output for the input values assigned to you.

Assignment of Input Values

Student Name	# rows (n)	# columns (m)	# coins (P)	Max value per coin (V)
Perry Butler	10	12	40	25
Latamla Culley-Triggs	10	12	35	35
Justin Epps	10	12	30	25
Kalil-Dan Ford	10	12	25	35
Chawne Harris	9	10	40	30
Ashly Horner	9	10	35	20
Martice Jackson	9	10	30	30
Jorian Lenard	9	10	25	20
Damian Patterson	8	10	30	40
Brandon Redmond	8	10	25	50
Daren Washington	8	10	40	35
Alicia Wells	8	10	35	30
Marcus Wynn	12	10	40	22
	12	10	35	28
	12	10	30	30
	12	10	25	32
	10	9	37	24

A sample screenshot of the execution of the program expected from you is shown below.

```
Enter the number of rows: 10
Enter the number of columns: 7
Enter the number of coins: 40
Enter the max. value for a coin: 30

Distribution of the Coin Values
25      16      4       6       0       18      0
26      0       0       7       0       3       0
0       0       0       26      2       11      21
5       0       0       0       17      0       0
0       4       0       14      0       0       17
0       20      27      0       12      1       26
0       26      29      22      6       0       0
0       13      14      8       0       23      0
0       20      0       0       24      25      12
13      0       12      27      0       28      27


Dynamic Programming Table
25      41      45      51      51      69      69
51      51      51      58      58      72      72
51      51      51      84      86      97      118
56      56      56      84      103     103     118
56      60      60      98      103     103     135
56      80      107     107     119     120     161
56      106     136     158     164     164     164
56      119     150     166     166     189     189
56      139     150     166     190     215     227
69      139     162     193     193     243     270


Path Traversed: [0 0, 1 0, 2 0, 3 0, 3 1, 4 1, 5 1, 5 2, 6 2, 6 3, 7 3, 7 4, 8 4
, 8 5, 9 5, 9 6]
```