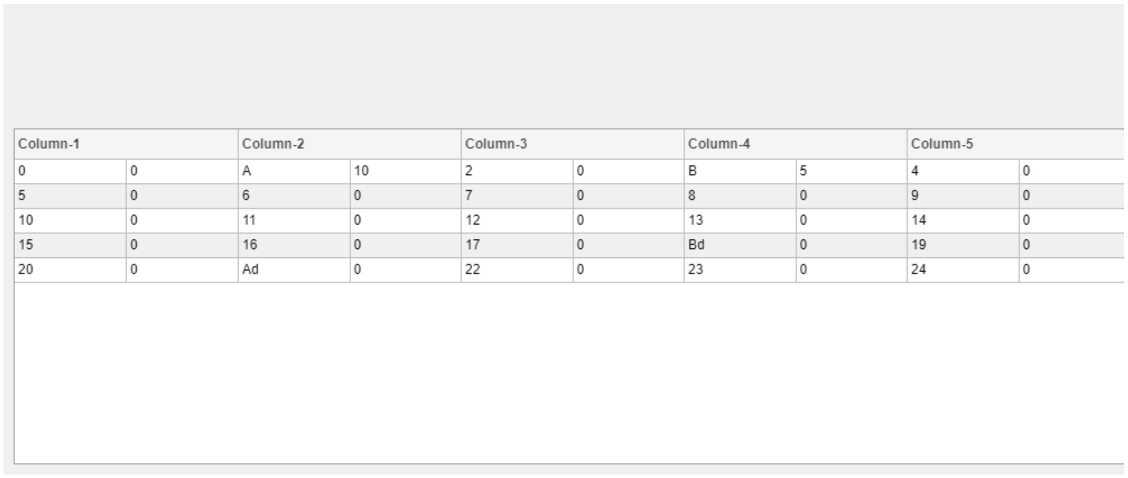
```
% Achyuth Nandikotkur
% V00975928
% Question #5

clear;
clc;

accuracyfactor = 0.1;
gamma = 0.8;

states = string(0:24);
statesWithPolicies = cell(5,5);
statevalues = zeros(1,24);

for i=1:numel(statesWithPolicies)
    if(i == 2)
        statesWithPolicies{i} = {'A', '####', 0, [0.25, 0.25, 0.25,
 0.25], 1};
    elseif(i == 4)
        statesWithPolicies{i} = {'B', '####', 0, [0.25, 0.25, 0.25,
 0.25], 1};
    elseif(i == 19)
        statesWithPolicies{i} = {'Bd', '####', 0, [0.25, 0.25, 0.25,
 0.25], 1};
    elseif(i == 22)
        statesWithPolicies{i} = {'Ad', '####', 0, [0.25, 0.25, 0.25,
 0.25], 1};
    else
        statesWithPolicies{i} = {states(i), '####', 0, [0.25, 0.25,
 0.25, 0.25], 1};
    end
end

innerloop = 1;
valueIterIndex = 1;

while innerloop
    tempstore = statesWithPolicies;
    delta = 0;
    for state=1:25
        lastStateValue = tempstore{state}{3};
        if(tempstore{state}{5} == 1)
            % left right up down
            intermediateValues = [0 0 0 0];

            % top side
            if(any(strcmp({'A'}, statesWithPolicies{state}{1})))
                % right
                intermediateValues(2) = (10 + gamma *
 statesWithPolicies{22}{3});

                % left
                intermediateValues(1) = (10 + gamma *
statesWithPolicies{22}{3});

                % up
                intermediateValues(3) = (10 + gamma *
statesWithPolicies{22}{3});

                % down
                intermediateValues(4) =  (10 + gamma *
statesWithPolicies{22}{3});
            elseif(any(strcmp({'B'}, statesWithPolicies{state}{1})))
                % right
                intermediateValues(2) = (5 + gamma *
statesWithPolicies{19}{3});

                % left
                intermediateValues(1) = (5 + gamma *
statesWithPolicies{19}{3});

                % up
                intermediateValues(3) = (5 + gamma *
statesWithPolicies{19}{3});

                % down
                intermediateValues(4) = (5 + gamma *
statesWithPolicies{19}{3});
            elseif(any(strcmp({'2'}, statesWithPolicies{state}{1})))
                % right
                intermediateValues(2) = (gamma *
statesWithPolicies{state+1}{3});

                % left
                intermediateValues(1) = (gamma *
statesWithPolicies{state-1}{3});

                % up
                intermediateValues(3) = (-1 + gamma *
statesWithPolicies{state}{3});

                % down
                intermediateValues(4) = (gamma *
statesWithPolicies{state+5}{3});
            % right side
            elseif(any(strcmp({'9', '14', '19'},
statesWithPolicies{state}{1})))
                % right
                intermediateValues(2) = (-1 + gamma *
statesWithPolicies{state}{3});

                % left
                intermediateValues(1) = (gamma *
statesWithPolicies{state-1}{3});

                % up
```

```
               intermediateValues(3) = (gamma *
statesWithPolicies{state-5}{3});

               % down
               intermediateValues(4) = (gamma *
statesWithPolicies{state+5}{3});
           % bottom side
           elseif(any(strcmp({'Ad', '22', '23'},
statesWithPolicies{state}{1})))
               % right
               intermediateValues(2) = (gamma *
statesWithPolicies{state+1}{3});

               % left
               intermediateValues(1) = (gamma *
statesWithPolicies{state-1}{3});

               % up
               intermediateValues(3) = (gamma *
statesWithPolicies{state-5}{3});

               % down
               intermediateValues(4) = (-1 + gamma *
statesWithPolicies{state}{3});
           % left side
           elseif(any(strcmp({'5', '10', '15'},
statesWithPolicies{state}{1})))
               % right
               intermediateValues(2) = (gamma *
statesWithPolicies{state+1}{3});

               % left
               intermediateValues(1) = (-1 + gamma *
statesWithPolicies{state}{3});

               % up
               intermediateValues(3) = (gamma *
statesWithPolicies{state-5}{3});

               % down
               intermediateValues(4) = (gamma *
statesWithPolicies{state+5}{3});
           % corners
           elseif('0' == statesWithPolicies{state}{1})
               % right
               intermediateValues(2) = (gamma *
statesWithPolicies{state+1}{3});

               % left
               intermediateValues(1) = (-1 + gamma *
statesWithPolicies{state}{3});

               % up
```

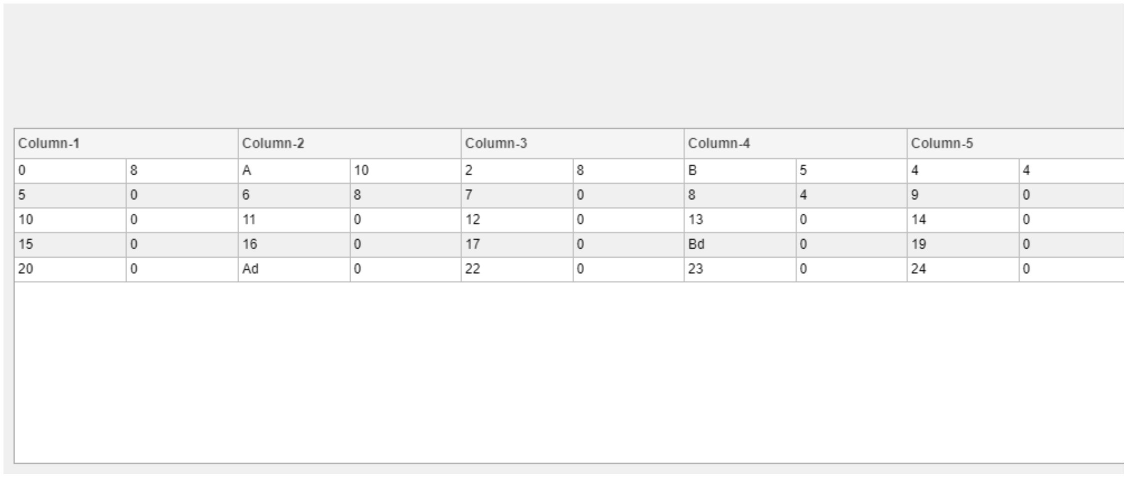
```
                intermediateValues(3) = (-1 + gamma *
statesWithPolicies{state}{3});

                % down
                intermediateValues(4) = (gamma *
statesWithPolicies{state+5}{3});
            elseif('4' == statesWithPolicies{state}{1})
                % right
                intermediateValues(2) = (-1 + gamma *
statesWithPolicies{state}{3});

                % left
                intermediateValues(1) = (gamma *
statesWithPolicies{state-1}{3});

                % up
                intermediateValues(3) = (-1 + gamma *
statesWithPolicies{state}{3});

                % down
                intermediateValues(4) = (gamma *
statesWithPolicies{state+5}{3});
            elseif('24' == statesWithPolicies{state}{1})
                % right
                intermediateValues(2) = (-1 + gamma *
statesWithPolicies{state}{3});

                % left
                intermediateValues(1) = (gamma *
statesWithPolicies{state-1}{3});

                % up
                intermediateValues(3) = (gamma *
statesWithPolicies{state-5}{3});

                % down
                intermediateValues(4) = (-1 + gamma *
statesWithPolicies{state}{3});
            elseif('20' == statesWithPolicies{state}{1})
                % right
                intermediateValues(2) = (gamma *
statesWithPolicies{state+1}{3});

                % left
                intermediateValues(1) = (-1 + gamma *
statesWithPolicies{state}{3});

                % up
                intermediateValues(3) = (gamma *
statesWithPolicies{state-5}{3});

                % down
                intermediateValues(4) = (-1 + gamma *
statesWithPolicies{state}{3});
```

```
            % All other cases
            else
                % right
                intermediateValues(2) = (gamma *
 statesWithPolicies{state+1}{3});

                % left
                intermediateValues(1) = (gamma *
 statesWithPolicies{state-1}{3});

                % up
                intermediateValues(3) = (gamma *
 statesWithPolicies{state-5}{3});

                % down
                intermediateValues(4) = (gamma *
 statesWithPolicies{state+5}{3});
            end

            maxval = max(intermediateValues);
            lia = ismember(intermediateValues, maxval);
            idx = find(lia);

            policy = '';
            equalityCheck = [0, 0, 0, 0];
            for i = 1:numel(idx)
                if(idx(i) == 1)
                    policy = policy + "#";
                end
                if(idx(i) == 2)
                    policy = policy + "#";
                end
                if(idx(i) == 3)
                    policy = policy + "#";
                end
                if(idx(i) == 4)
                    policy = policy + "#";
                end
            end

            tempstore{state}{2} = policy;
            tempstore{state}{3} = maxval;

            delta = max(delta, abs(lastStateValue - tempstore{state}
{3}));
        end
    end

    statesWithPolicies = tempstore;
    if(delta < accuracyfactor)
        innerloop = 0;
    end
    valueIterIndex = valueIterIndex + 1;
    printPolicy(statesWithPolicies, valueIterIndex-1, 0)
```

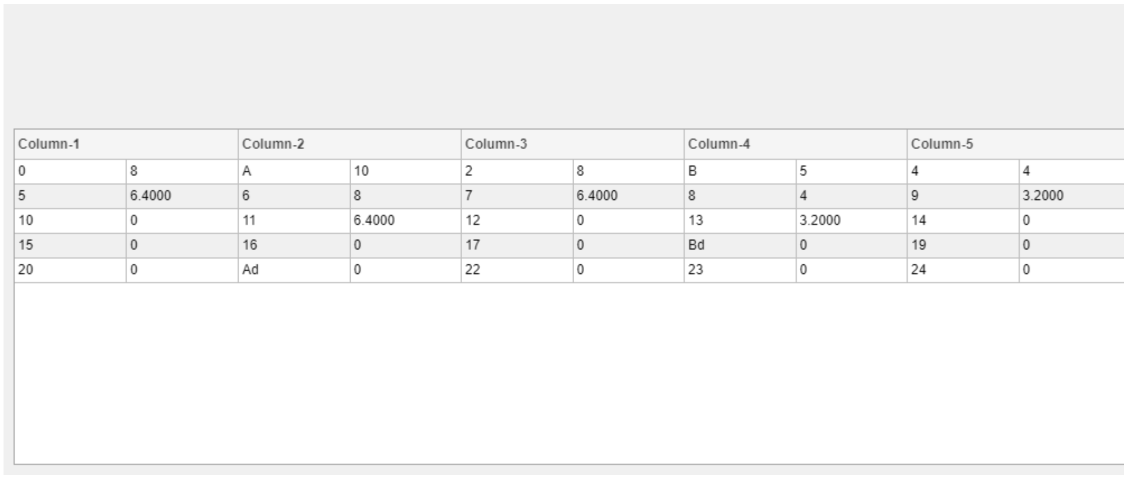
```
end

lastComputation(statesWithPolicies, gamma);

function lastComputation(statesWithPolicies, gamma)
    tempstore = statesWithPolicies;
    for state=1:25
        % left right up down
        intermediateValues = [0 0 0 0];

        % top side
        if(any(strcmp({'A'}, statesWithPolicies{state}{1})))
            % right
            intermediateValues(2) = (10 + gamma *
 statesWithPolicies{22}{3});

            % left
            intermediateValues(1) = (10 + gamma *
 statesWithPolicies{22}{3});

            % up
            intermediateValues(3) = (10 + gamma *
 statesWithPolicies{22}{3});

            % down
            intermediateValues(4) =  (10 + gamma *
 statesWithPolicies{22}{3});
        elseif(any(strcmp({'B'}, statesWithPolicies{state}{1})))
            % right
            intermediateValues(2) = (5 + gamma *
 statesWithPolicies{19}{3});

            % left
            intermediateValues(1) = (5 + gamma *
 statesWithPolicies{19}{3});

            % up
            intermediateValues(3) = (5 + gamma *
 statesWithPolicies{19}{3});

            % down
            intermediateValues(4) = (5 + gamma *
 statesWithPolicies{19}{3});
        elseif(any(strcmp({'2'}, statesWithPolicies{state}{1})))
            % right
            intermediateValues(2) = (gamma * statesWithPolicies{state
+1}{3});

            % left
            intermediateValues(1) = (gamma *
 statesWithPolicies{state-1}{3});

            % up
            intermediateValues(3) = (-1 + gamma *
 statesWithPolicies{state}{3});

            % down
            intermediateValues(4) = (gamma * statesWithPolicies{state
+5}{3});
        % right side
        elseif(any(strcmp({'9', '14', '19'}, statesWithPolicies{state}
{1})))
            % right
            intermediateValues(2) = (-1 + gamma *
 statesWithPolicies{state}{3});

            % left
            intermediateValues(1) = (0 + gamma *
 statesWithPolicies{state-1}{3});

            % up
            intermediateValues(3) = (0 + gamma *
 statesWithPolicies{state-5}{3});

            % down
            intermediateValues(4) = (0 + gamma *
 statesWithPolicies{state+5}{3});
        % bottom side
        elseif(any(strcmp({'Ad', '22', '23'},
 statesWithPolicies{state}{1})))
            % right
            intermediateValues(2) = (0 + gamma *
 statesWithPolicies{state+1}{3});

            % left
            intermediateValues(1) = (0 + gamma *
 statesWithPolicies{state-1}{3});

            % up
            intermediateValues(3) = (0 + gamma *
 statesWithPolicies{state-5}{3});

            % down
            intermediateValues(4) = (-1 + gamma *
 statesWithPolicies{state}{3});
        % left side
        elseif(any(strcmp({'5', '10', '15'}, statesWithPolicies{state}
{1})))
            % right
            intermediateValues(2) = (0 + gamma *
 statesWithPolicies{state+1}{3});

            % left
            intermediateValues(1) = (-1 + gamma *
 statesWithPolicies{state}{3});

            % up
            intermediateValues(3) = (0 + gamma *
statesWithPolicies{state-5}{3});

            % down
            intermediateValues(4) = (0 + gamma *
statesWithPolicies{state+5}{3});
        % corners
        elseif('0' == statesWithPolicies{state}{1})
            % right
            intermediateValues(2) = (0 + gamma *
statesWithPolicies{state+1}{3});

            % left
            intermediateValues(1) = (-1 + gamma *
statesWithPolicies{state}{3});

            % up
            intermediateValues(3) = (-1 + gamma *
statesWithPolicies{state}{3});

            % down
            intermediateValues(4) = (0 + gamma *
statesWithPolicies{state+5}{3});
        elseif('4' == statesWithPolicies{state}{1})
            % right
            intermediateValues(2) = (-1 + gamma *
statesWithPolicies{state}{3});

            % left
            intermediateValues(1) = (0 + gamma *
statesWithPolicies{state-1}{3});

            % up
            intermediateValues(3) = (-1 + gamma *
statesWithPolicies{state}{3});

            % down
            intermediateValues(4) = (0 + gamma *
statesWithPolicies{state+5}{3});
        elseif('24' == statesWithPolicies{state}{1})
            % right
            intermediateValues(2) = (-1 + gamma *
statesWithPolicies{state}{3});

            % left
            intermediateValues(1) = (0 + gamma *
statesWithPolicies{state-1}{3});

            % up
            intermediateValues(3) = (0 + gamma *
statesWithPolicies{state-5}{3});

            % down
            intermediateValues(4) = (-1 + gamma * 
statesWithPolicies{state}{3});
        elseif('20' == statesWithPolicies{state}{1})
            % right
            intermediateValues(2) = (0 + gamma * 
statesWithPolicies{state+1}{3});

            % left
            intermediateValues(1) = (-1 + gamma * 
statesWithPolicies{state}{3});

            % up
            intermediateValues(3) = (0 + gamma * 
statesWithPolicies{state-5}{3});

            % down
            intermediateValues(4) = (-1 + gamma * 
statesWithPolicies{state}{3});
        % All other cases
        else
            % right
            intermediateValues(2) = (0 + gamma * 
statesWithPolicies{state+1}{3});

            % left
            intermediateValues(1) = (0 + gamma * 
statesWithPolicies{state-1}{3});

            % up
            intermediateValues(3) = (0 + gamma * 
statesWithPolicies{state-5}{3});

            % down
            intermediateValues(4) = (0 + gamma * 
statesWithPolicies{state+5}{3});
        end

        maxval = max(intermediateValues);
        lia = ismember(intermediateValues, maxval);
        idx = find(lia);

        policy = '';
        for i = 1:numel(idx)
            if(idx(i) == 1)
                policy = policy + "#";
            end
            if(idx(i) == 2)
                policy = policy + "#";
            end
            if(idx(i) == 3)
                policy = policy + "#";
            end
            if(idx(i) == 4)
                policy = policy + "#";
            end
        end
        tempstore{state}{2} = policy;
    end
    fprintf('\n\n')

    printPolicy(tempstore, 0, 1)
end

function printPolicy(statesWithPolicies, policyNumber, optimal)
    temporary = statesWithPolicies;
    for final = 1:25
        if(optimal == 0)
            temporary{final}(2) = [];
            temporary{final}(3) = [];
            temporary{final}(3) = [];
        else
            temporary{final}(4) = [];
            temporary{final}(4) = [];
        end
    end
    t = cell2table(transpose(temporary),'VariableNames',
{'Column-1', 'Column-2', 'Column-3', 'Column-4', 'Column-5'});
    fig = uifigure;
    if(optimal == 0)
        fig.Name = ['Value Iteration: ',  num2str(policyNumber)];
    else
        fig.Name = 'Optimal Value & Policy';
    end
    fig.Position(3) = 1000;

    uitable(fig,'Data',t, 'ColumnWidth',{199, 199,  199, 199,
 199}, 'Position',[10 10 1000 300]);
end
```



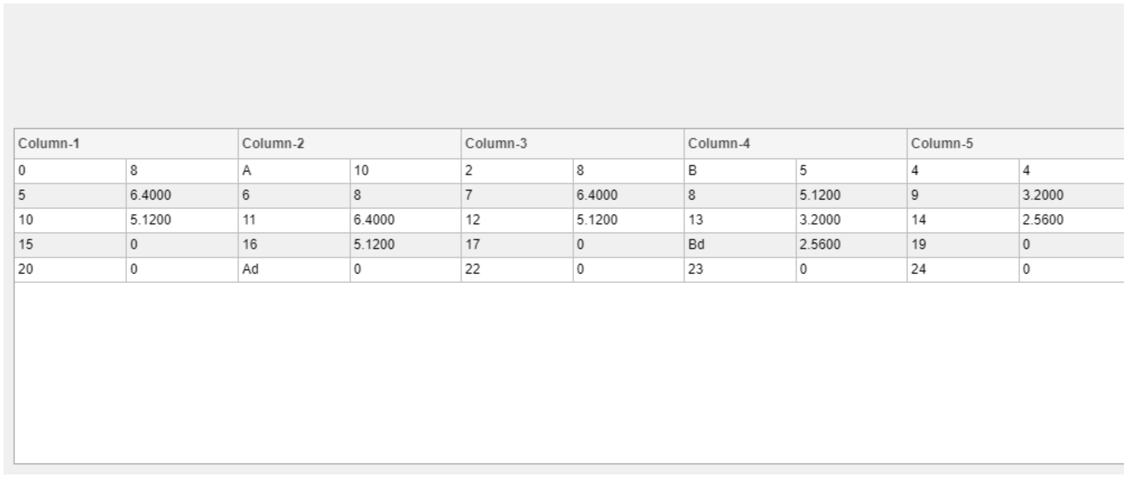
Column-1		Column-2		Column-3		Column-4		Column-5	
0	0	A	10	2	0	B	5	4	0
5	0	6	0	7	0	8	0	9	0
10	0	11	0	12	0	13	0	14	0
15	0	16	0	17	0	Bd	0	19	0
20	0	Ad	0	22	0	23	0	24	0



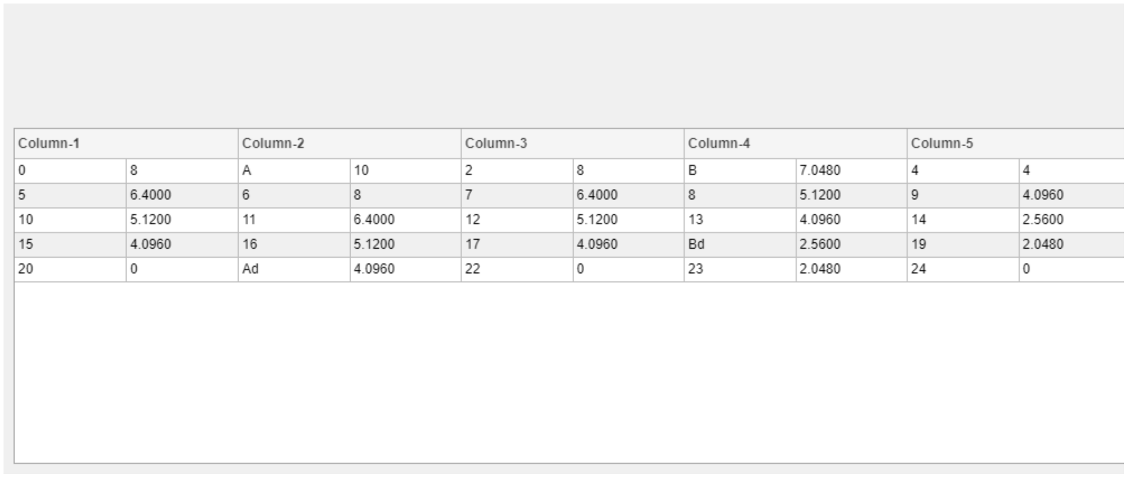
Column-1		Column-2		Column-3		Column-4		Column-5	
0	8	A	10	2	8	B	5	4	4
5	0	6	8	7	0	8	4	9	0
10	0	11	0	12	0	13	0	14	0
15	0	16	0	17	0	Bd	0	19	0
20	0	Ad	0	22	0	23	0	24	0



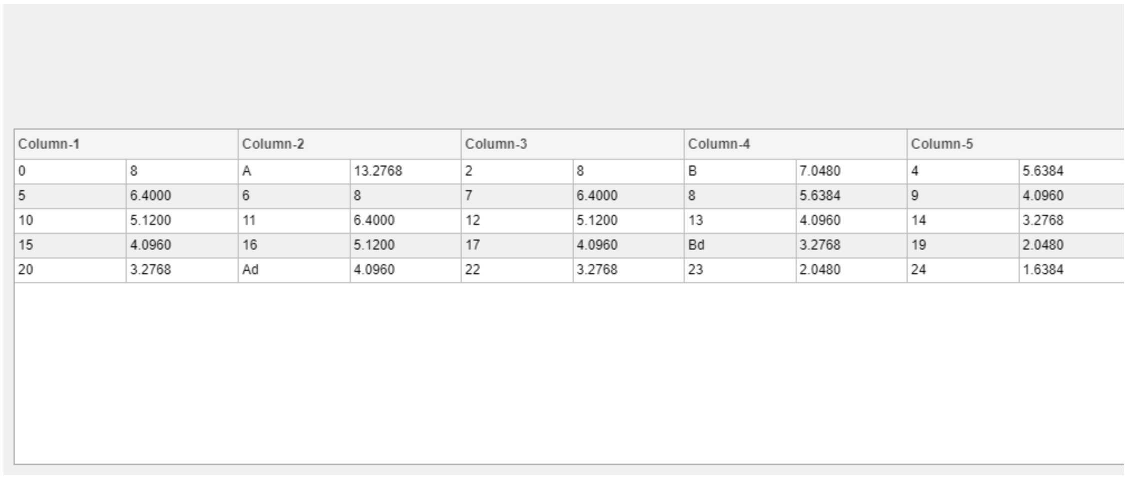
Column-1		Column-2		Column-3		Column-4		Column-5	
0	8	A	10	2	8	B	5	4	4
5	6.4000	6	8	7	6.4000	8	4	9	3.2000
10	0	11	6.4000	12	0	13	3.2000	14	0
15	0	16	0	17	0	Bd	0	19	0
20	0	Ad	0	22	0	23	0	24	0



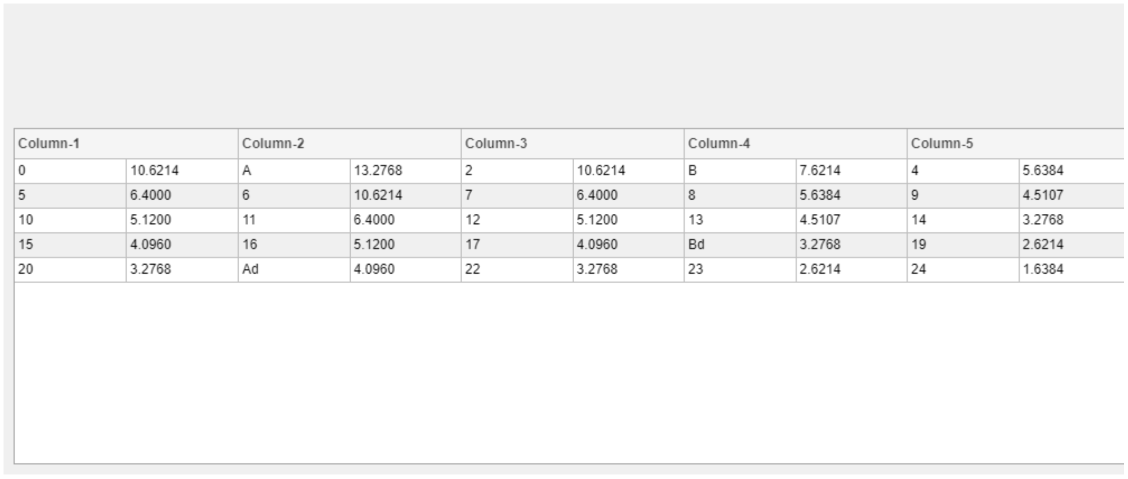
Column-1		Column-2		Column-3		Column-4		Column-5	
0	8	A	10	2	8	B	5	4	4
5	6.4000	6	8	7	6.4000	8	5.1200	9	3.2000
10	5.1200	11	6.4000	12	5.1200	13	3.2000	14	2.5600
15	0	16	5.1200	17	0	Bd	2.5600	19	0
20	0	Ad	0	22	0	23	0	24	0



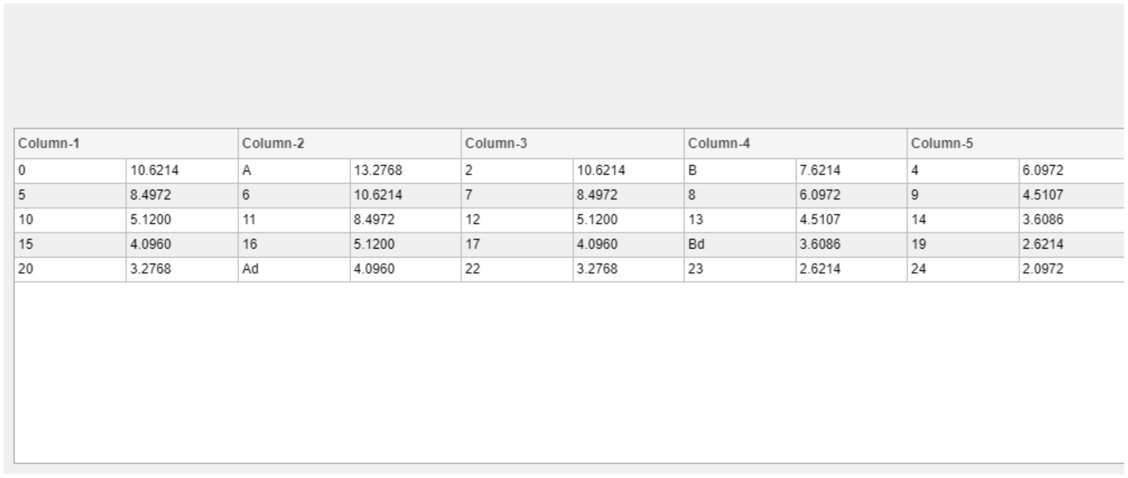
Column-1		Column-2		Column-3		Column-4		Column-5	
0	8	A	10	2	8	B	7.0480	4	4
5	6.4000	6	8	7	6.4000	8	5.1200	9	4.0960
10	5.1200	11	6.4000	12	5.1200	13	4.0960	14	2.5600
15	4.0960	16	5.1200	17	4.0960	Bd	2.5600	19	2.0480
20	0	Ad	4.0960	22	0	23	2.0480	24	0



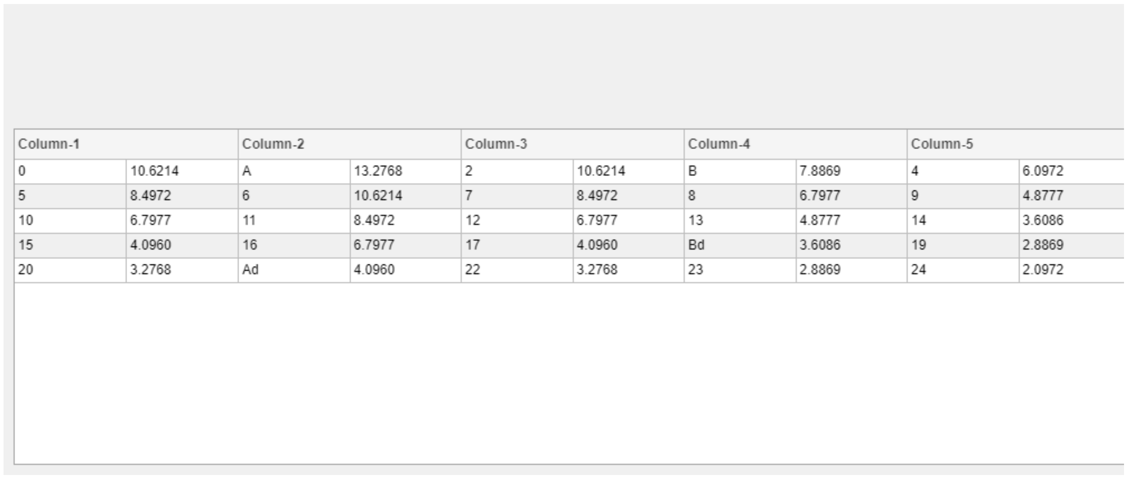
Column-1		Column-2		Column-3		Column-4		Column-5	
0	8	A	13.2768	2	8	B	7.0480	4	5.6384
5	6.4000	6	8	7	6.4000	8	5.6384	9	4.0960
10	5.1200	11	6.4000	12	5.1200	13	4.0960	14	3.2768
15	4.0960	16	5.1200	17	4.0960	Bd	3.2768	19	2.0480
20	3.2768	Ad	4.0960	22	3.2768	23	2.0480	24	1.6384



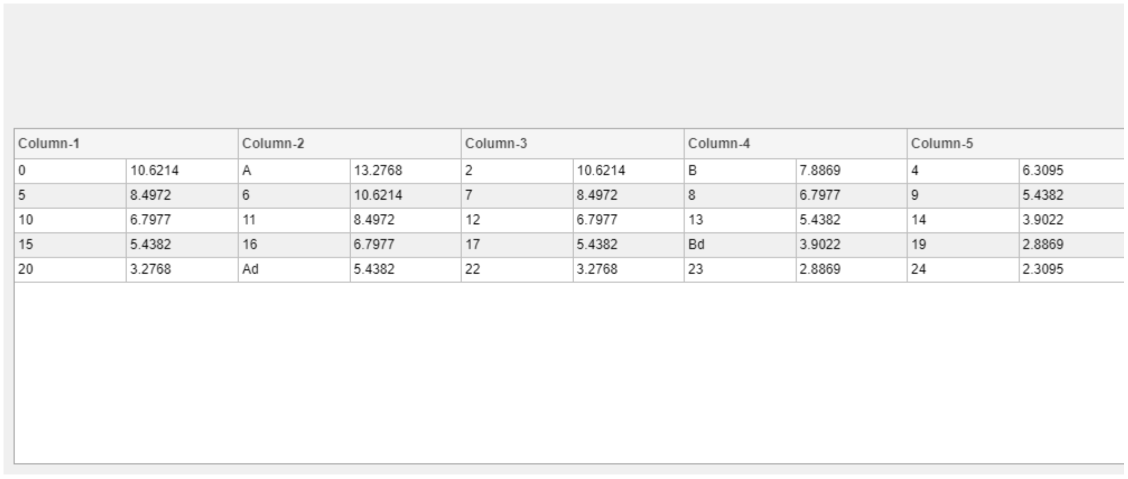
Column-1		Column-2		Column-3		Column-4		Column-5	
0	10.6214	A	13.2768	2	10.6214	B	7.6214	4	5.6384
5	6.4000	6	10.6214	7	6.4000	8	5.6384	9	4.5107
10	5.1200	11	6.4000	12	5.1200	13	4.5107	14	3.2768
15	4.0960	16	5.1200	17	4.0960	Bd	3.2768	19	2.6214
20	3.2768	Ad	4.0960	22	3.2768	23	2.6214	24	1.6384



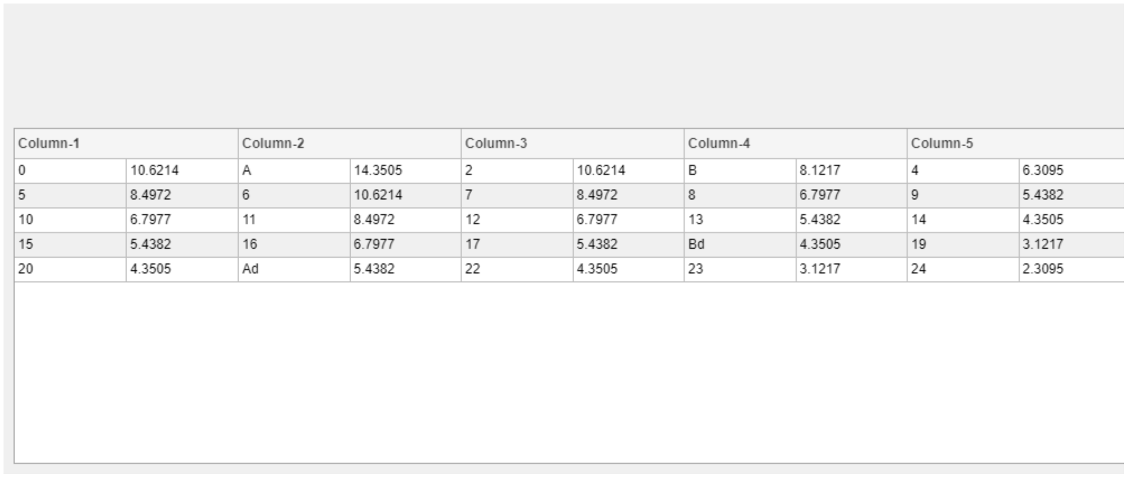
Column-1		Column-2		Column-3		Column-4		Column-5	
0	10.6214	A	13.2768	2	10.6214	B	7.6214	4	6.0972
5	8.4972	6	10.6214	7	8.4972	8	6.0972	9	4.5107
10	5.1200	11	8.4972	12	5.1200	13	4.5107	14	3.6086
15	4.0960	16	5.1200	17	4.0960	Bd	3.6086	19	2.6214
20	3.2768	Ad	4.0960	22	3.2768	23	2.6214	24	2.0972



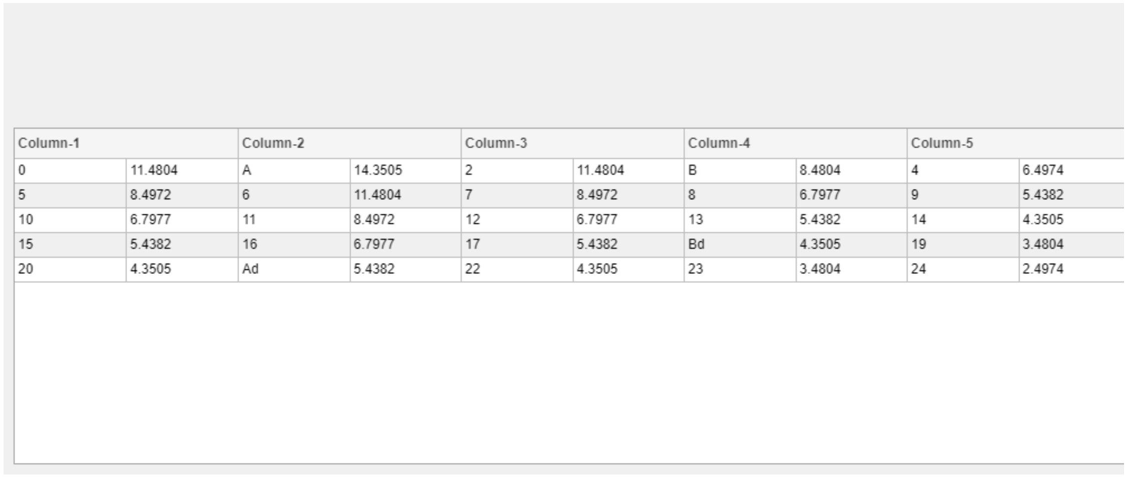
Column-1		Column-2		Column-3		Column-4		Column-5	
0	10.6214	A	13.2768	2	10.6214	B	7.8869	4	6.0972
5	8.4972	6	10.6214	7	8.4972	8	6.7977	9	4.8777
10	6.7977	11	8.4972	12	6.7977	13	4.8777	14	3.6086
15	4.0960	16	6.7977	17	4.0960	Bd	3.6086	19	2.8869
20	3.2768	Ad	4.0960	22	3.2768	23	2.8869	24	2.0972



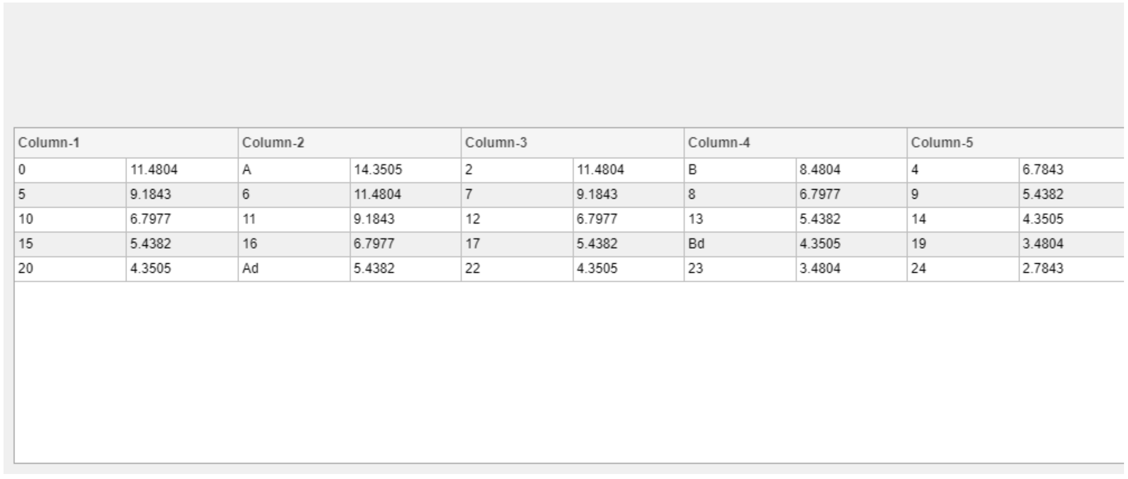
Column-1		Column-2		Column-3		Column-4		Column-5	
0	10.6214	A	13.2768	2	10.6214	B	7.8869	4	6.3095
5	8.4972	6	10.6214	7	8.4972	8	6.7977	9	5.4382
10	6.7977	11	8.4972	12	6.7977	13	5.4382	14	3.9022
15	5.4382	16	6.7977	17	5.4382	Bd	3.9022	19	2.8869
20	3.2768	Ad	5.4382	22	3.2768	23	2.8869	24	2.3095



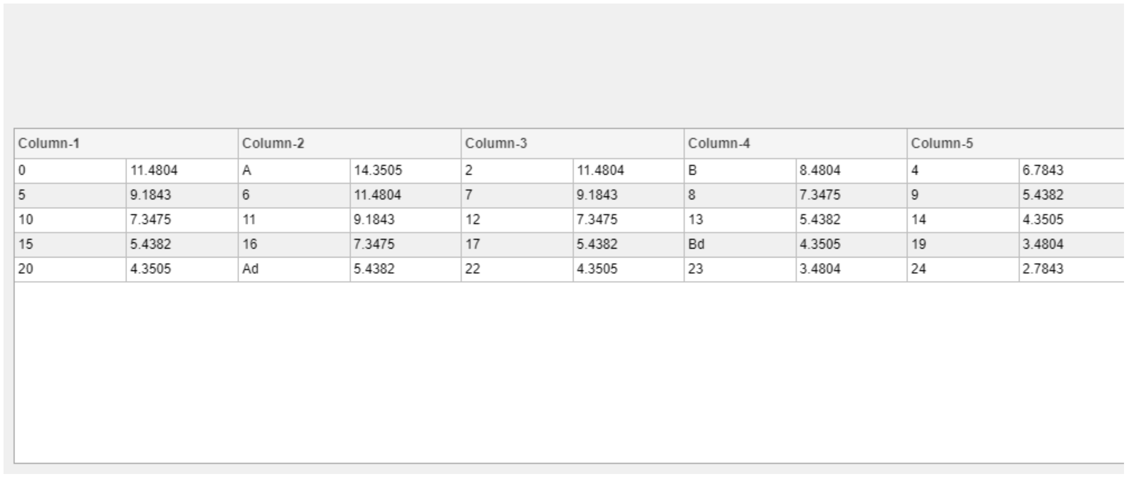
Column-1		Column-2		Column-3		Column-4		Column-5	
0	10.6214	A	14.3505	2	10.6214	B	8.1217	4	6.3095
5	8.4972	6	10.6214	7	8.4972	8	6.7977	9	5.4382
10	6.7977	11	8.4972	12	6.7977	13	5.4382	14	4.3505
15	5.4382	16	6.7977	17	5.4382	Bd	4.3505	19	3.1217
20	4.3505	Ad	5.4382	22	4.3505	23	3.1217	24	2.3095



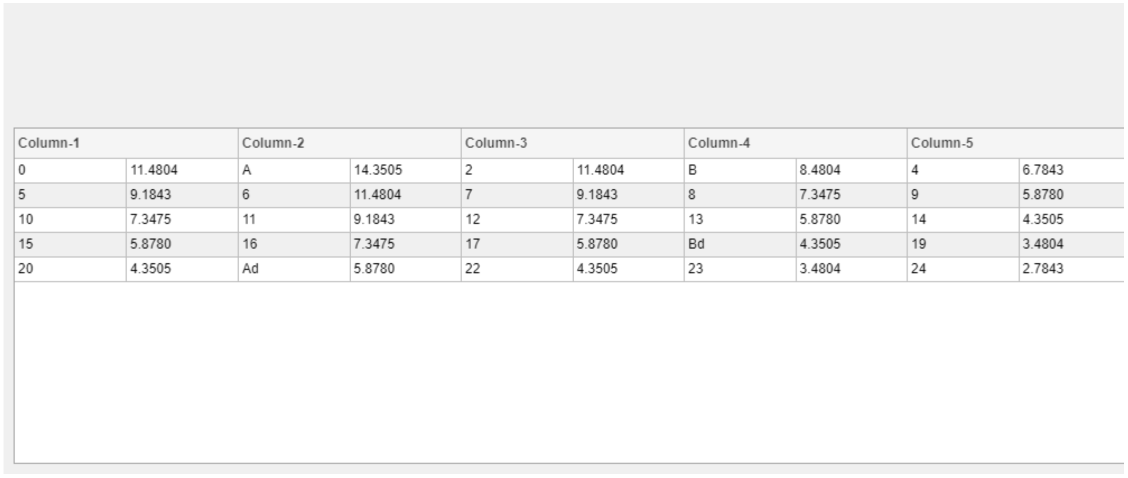
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.4804	A	14.3505	2	11.4804	B	8.4804	4	6.4974
5	8.4972	6	11.4804	7	8.4972	8	6.7977	9	5.4382
10	6.7977	11	8.4972	12	6.7977	13	5.4382	14	4.3505
15	5.4382	16	6.7977	17	5.4382	Bd	4.3505	19	3.4804
20	4.3505	Ad	5.4382	22	4.3505	23	3.4804	24	2.4974



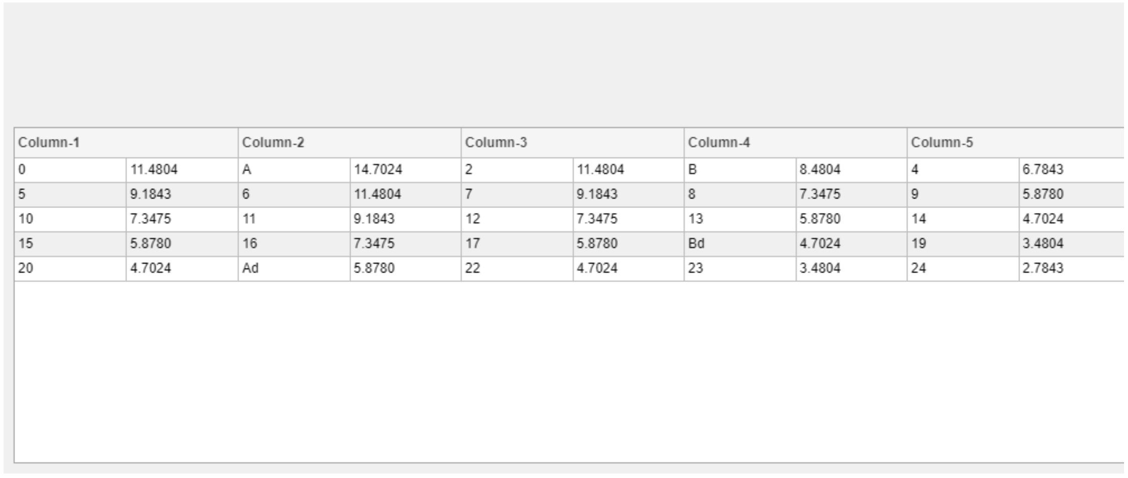
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.4804	A	14.3505	2	11.4804	B	8.4804	4	6.7843
5	9.1843	6	11.4804	7	9.1843	8	6.7977	9	5.4382
10	6.7977	11	9.1843	12	6.7977	13	5.4382	14	4.3505
15	5.4382	16	6.7977	17	5.4382	Bd	4.3505	19	3.4804
20	4.3505	Ad	5.4382	22	4.3505	23	3.4804	24	2.7843



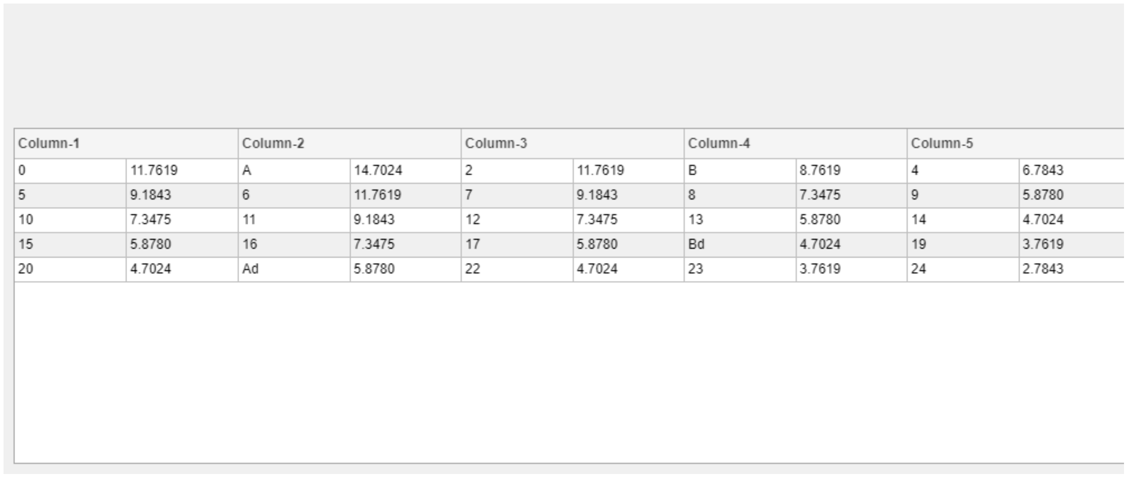
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.4804	A	14.3505	2	11.4804	B	8.4804	4	6.7843
5	9.1843	6	11.4804	7	9.1843	8	7.3475	9	5.4382
10	7.3475	11	9.1843	12	7.3475	13	5.4382	14	4.3505
15	5.4382	16	7.3475	17	5.4382	Bd	4.3505	19	3.4804
20	4.3505	Ad	5.4382	22	4.3505	23	3.4804	24	2.7843



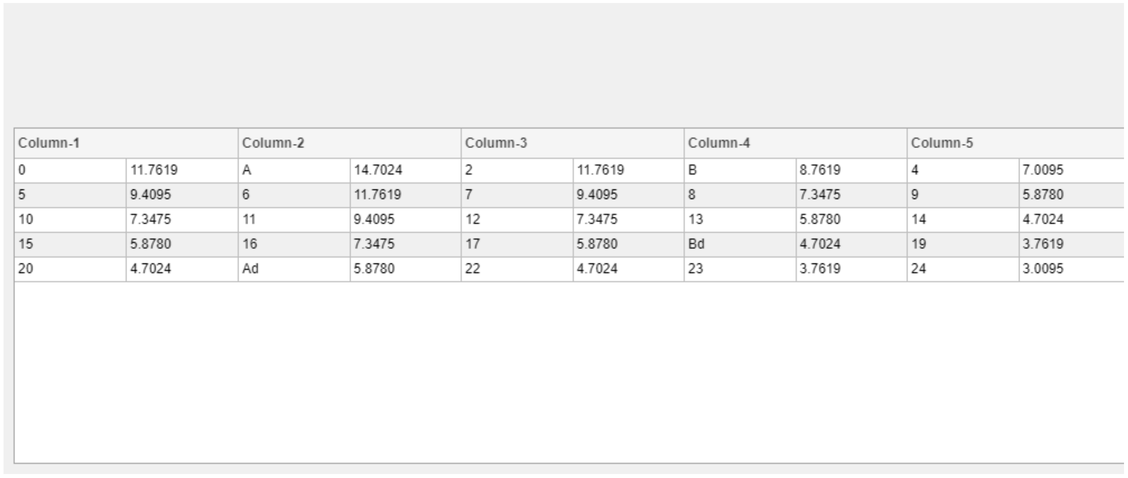
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.4804	A	14.3505	2	11.4804	B	8.4804	4	6.7843
5	9.1843	6	11.4804	7	9.1843	8	7.3475	9	5.8780
10	7.3475	11	9.1843	12	7.3475	13	5.8780	14	4.3505
15	5.8780	16	7.3475	17	5.8780	Bd	4.3505	19	3.4804
20	4.3505	Ad	5.8780	22	4.3505	23	3.4804	24	2.7843



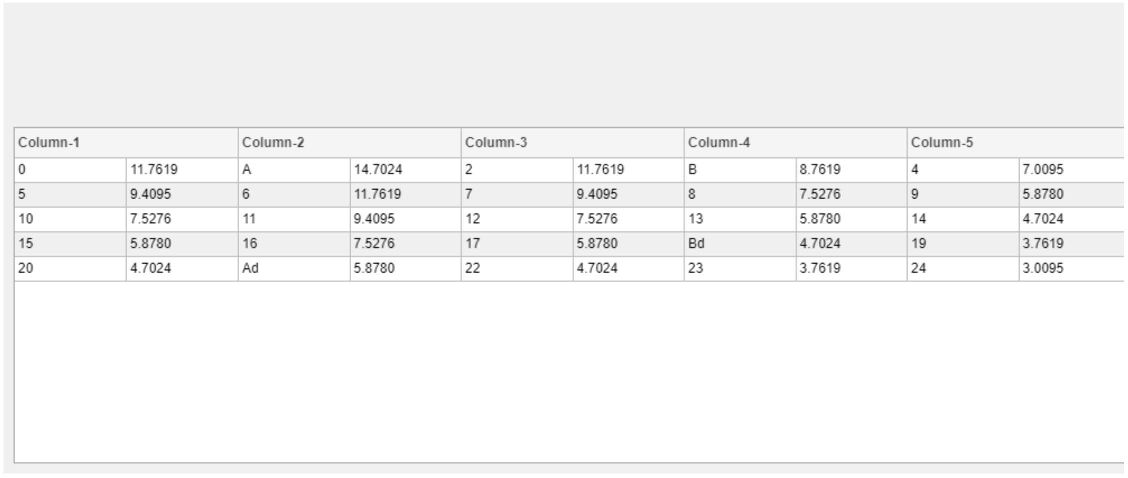
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.4804	A	14.7024	2	11.4804	B	8.4804	4	6.7843
5	9.1843	6	11.4804	7	9.1843	8	7.3475	9	5.8780
10	7.3475	11	9.1843	12	7.3475	13	5.8780	14	4.7024
15	5.8780	16	7.3475	17	5.8780	Bd	4.7024	19	3.4804
20	4.7024	Ad	5.8780	22	4.7024	23	3.4804	24	2.7843



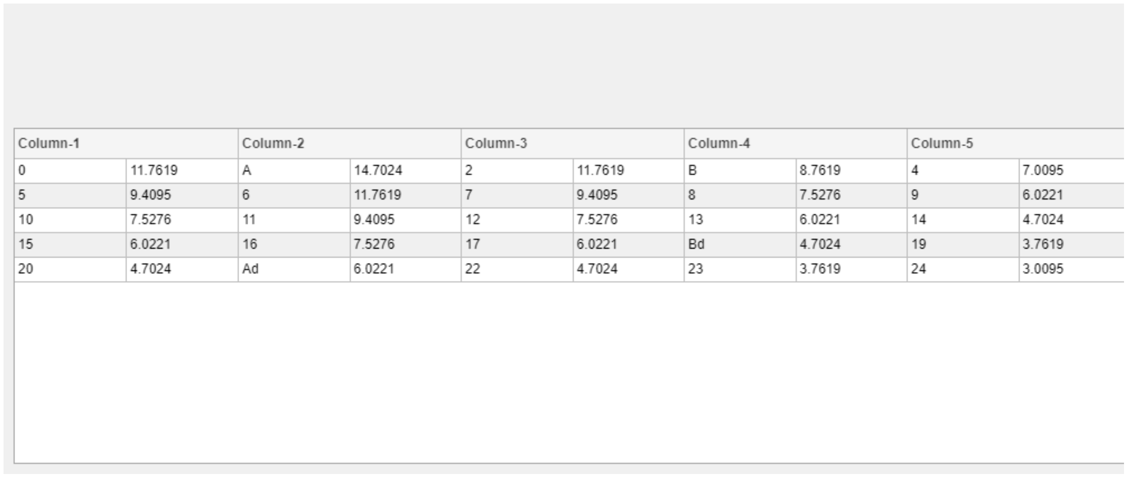
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.7619	A	14.7024	2	11.7619	B	8.7619	4	6.7843
5	9.1843	6	11.7619	7	9.1843	8	7.3475	9	5.8780
10	7.3475	11	9.1843	12	7.3475	13	5.8780	14	4.7024
15	5.8780	16	7.3475	17	5.8780	Bd	4.7024	19	3.7619
20	4.7024	Ad	5.8780	22	4.7024	23	3.7619	24	2.7843



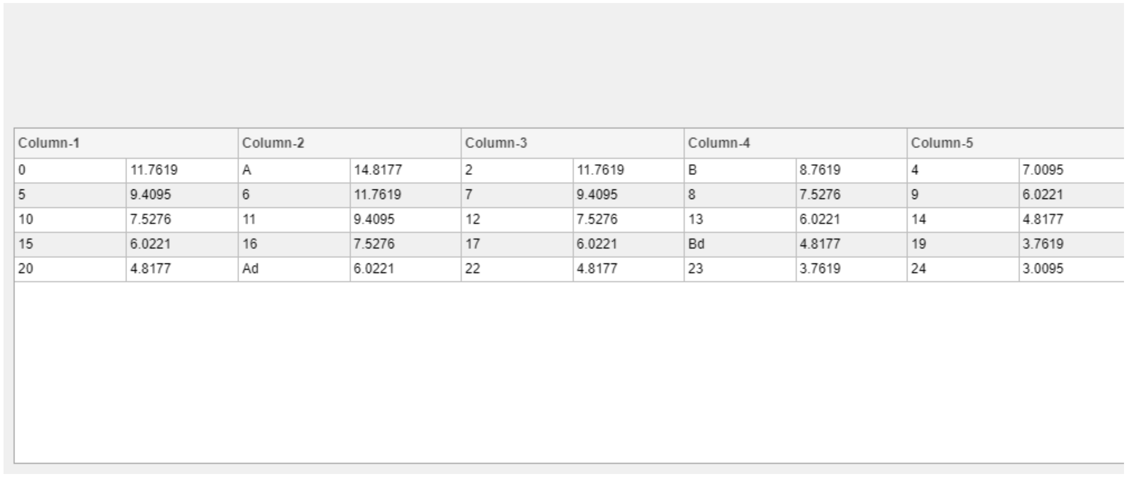
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.7619	A	14.7024	2	11.7619	B	8.7619	4	7.0095
5	9.4095	6	11.7619	7	9.4095	8	7.3475	9	5.8780
10	7.3475	11	9.4095	12	7.3475	13	5.8780	14	4.7024
15	5.8780	16	7.3475	17	5.8780	Bd	4.7024	19	3.7619
20	4.7024	Ad	5.8780	22	4.7024	23	3.7619	24	3.0095



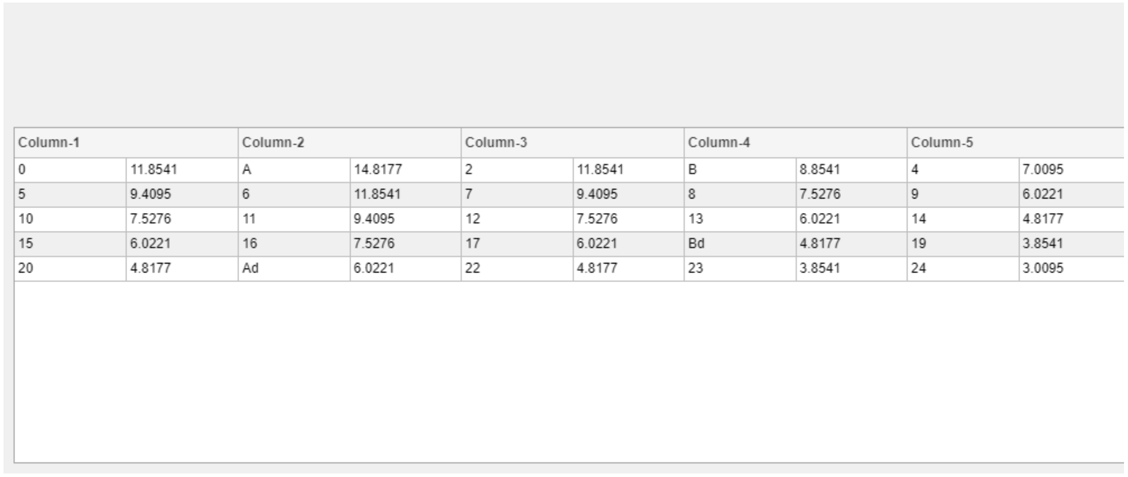
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.7619	A	14.7024	2	11.7619	B	8.7619	4	7.0095
5	9.4095	6	11.7619	7	9.4095	8	7.5276	9	5.8780
10	7.5276	11	9.4095	12	7.5276	13	5.8780	14	4.7024
15	5.8780	16	7.5276	17	5.8780	Bd	4.7024	19	3.7619
20	4.7024	Ad	5.8780	22	4.7024	23	3.7619	24	3.0095



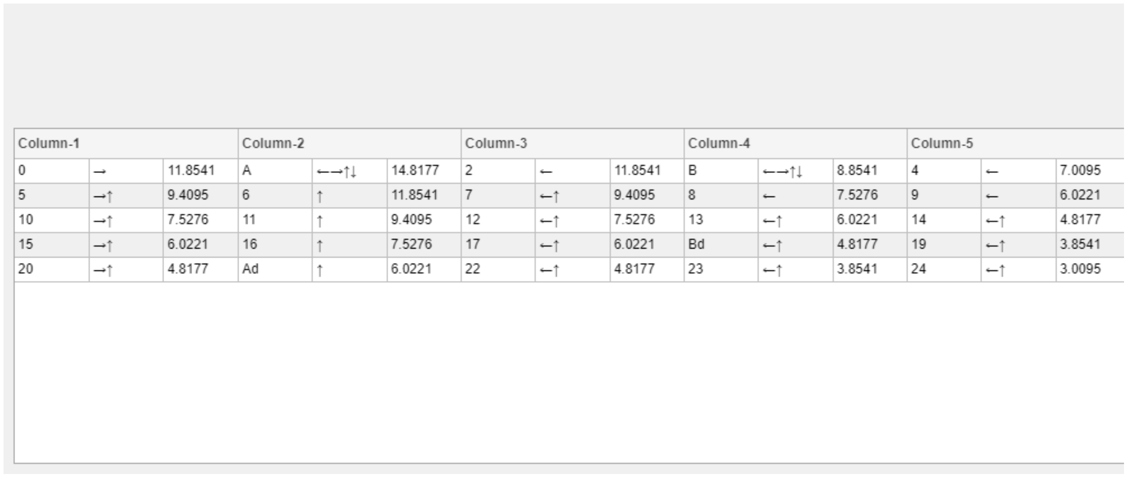
Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.7619	A	14.7024	2	11.7619	B	8.7619	4	7.0095
5	9.4095	6	11.7619	7	9.4095	8	7.5276	9	6.0221
10	7.5276	11	9.4095	12	7.5276	13	6.0221	14	4.7024
15	6.0221	16	7.5276	17	6.0221	Bd	4.7024	19	3.7619
20	4.7024	Ad	6.0221	22	4.7024	23	3.7619	24	3.0095



Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.7619	A	14.8177	2	11.7619	B	8.7619	4	7.0095
5	9.4095	6	11.7619	7	9.4095	8	7.5276	9	6.0221
10	7.5276	11	9.4095	12	7.5276	13	6.0221	14	4.8177
15	6.0221	16	7.5276	17	6.0221	Bd	4.8177	19	3.7619
20	4.8177	Ad	6.0221	22	4.8177	23	3.7619	24	3.0095



Column-1		Column-2		Column-3		Column-4		Column-5	
0	11.8541	A	14.8177	2	11.8541	B	8.8541	4	7.0095
5	9.4095	6	11.8541	7	9.4095	8	7.5276	9	6.0221
10	7.5276	11	9.4095	12	7.5276	13	6.0221	14	4.8177
15	6.0221	16	7.5276	17	6.0221	Bd	4.8177	19	3.8541
20	4.8177	Ad	6.0221	22	4.8177	23	3.8541	24	3.0095



Column-1			Column-2			Column-3			Column-4			Column-5		
0	→	11.8541	A	←→↑↓	14.8177	2	←	11.8541	B	←→↑↓	8.8541	4	←	7.0095
5	→↑	9.4095	6	↑	11.8541	7	←↑	9.4095	8	←	7.5276	9	←	6.0221
10	→↑	7.5276	11	↑	9.4095	12	←↑	7.5276	13	←↑	6.0221	14	←↑	4.8177
15	→↑	6.0221	16	↑	7.5276	17	←↑	6.0221	Bd	←↑	4.8177	19	←↑	3.8541
20	→↑	4.8177	Ad	↑	6.0221	22	←↑	4.8177	23	←↑	3.8541	24	←↑	3.0095

Published with MATLAB® R2021a