```python
from program2_1 import Dvector

M = 2 # 行の要素数
N = 3 # 列の要素数

class Dmatrix(object):
    def __init__(self, row_head_idx: int, row_last_idx: int, col_head_idx: int, col_last_idx: int):
        matrix = Dvector(row_head_idx, row_last_idx)
        for i in range(row_head_idx, row_last_idx+1):
            matrix[i] = Dvector(col_head_idx, col_last_idx)
        self.matrix = matrix
        self.row_head_idx = row_head_idx
        self.row_last_idx = row_last_idx
        self.col_head_idx = col_head_idx
        self.col_last_idx = col_last_idx

    def __repr__(self):
        return "[" + "\n ".join(map(str, self.matrix)) + "]"

    def __iter__(self):
        return iter(self.matrix)

    def __getitem__(self, key):
        return self.matrix[key]

    def __setitem__(self, key, value):
        self.matrix[key] = value

    # オブジェクトの値渡しで使用
    def copy(self):
        cpy = Dmatrix(self.row_head_idx, self.row_last_idx, self.col_head_idx, self.col_last_idx)
        for i in range(self.row_head_idx, self.row_last_idx+1):
            cpy.matrix[i] = self.matrix[i].copy()
        return cpy


def main():
    global M, N
    a = Dmatrix(1,M,1,N)

    for i in range(1, M+1):
        for j in range(1, N+1):
            a[i][j] = (i+j) / 2.0
            print(f"a[{i}][{j}]={a[i][j]}")


if __name__ == "__main__":
    main()
```