```go
// 後で他のソースコードから関数をモジュールとして参照するので、パッケージをexportしている。
// Go言語では、初めの文字が大文字の関数が公開され、小文字の関数は公開されない。
package locust

import (
    "../matmul"    // MatVec関数, MatMlt関数, Transpose関数, 定数N
    "encoding/csv" // 計算結果をCSVに出力するため
    "fmt"          // Sprint関数を使うため
    "os"           // 計算結果をCSVに出力するため
)

// #define N 3 の代わり。
const N = matmul.N

func main() {
    // c = 0.5, s = 0.05の場合の、各時刻における各地点の、バッタGの存在確率を計算している
    res1 := Calculation(0.5, 0.05)
    // c = 0.5, s = 0.15の場合の、各時刻における各地点の、バッタGの存在確率を計算している
    res2 := Calculation(0.5, 0.15)
    // c = 0.5, s = 0.5の場合の、各時刻における各地点の、バッタGの存在確率を計算している
    res3 := Calculation(0.5, 0.5)

    // 計算結果をCSVに出力している
    WriteCSV("0.05", res1)
    WriteCSV("0.15", res2)
    WriteCSV("0.5", res3)
}

// 書き出すファイル名と、書き出す内容を含んだArrayを受け取り、
// CSVファイルに書き出す関数
func WriteCSV(filename string, resVec [61][N]float64) {
    // CSVを書き出すファイルを指定
    file, _ := os.Create("res/" + filename + ".csv")
    writer := csv.NewWriter(file)
    // CSVのヘッダーを指定
    writer.Write([]string{
        "t",
        "point 0",
        "point 1",
        "point 2",
        "point 3",
        "point 4",
        "point 5",
    })
    for t := 0; t <= 60; t++ {
        // writer.Write 関数はstringのArrayしか受け取らないので、
        // resVec[t]の各要素をfmt.Sprint関数でstringに変換して、
        // そのArrayを生成し、それをwriter.Writeに渡している。
        writer.Write(
            []string{
                fmt.Sprint(t),
                fmt.Sprint(resVec[t][0]),
                fmt.Sprint(resVec[t][1]),
                fmt.Sprint(resVec[t][2]),
                fmt.Sprint(resVec[t][3]),
                fmt.Sprint(resVec[t][4]),
                fmt.Sprint(resVec[t][5]),
            },
        )
    }
    // バッファをファイルへ出力する
    writer.Flush()
}

// パラメーターcとsを受け取り、0 <= t <= 60の範囲のバッタGの
// 各地点での存在確率を計算する
func Calculation(c float64, s float64) [61][N]float64 {
    // 移動確率を表す行列を計算し、取得する
    var probMat [N][N]float64 = getProbMat(c, s)

    var resultVectors [61][N]float64                // resultVectors[t]: 時刻tに於ける、Gの地点0~5での存在確率を表すベクトル
    resultVectors[0] = [N]float64{1, 0, 0, 0, 0, 0} // t = 0のとき、バッタGの地点0~5での存在確率は
```

```
1,0,0,0,0,0である
    // t = 1, 2, ... , 60について、
    // Gの存在確率ベクトルを求める
    for t := 1; t <= 60; t++ {
        // ある時刻tのGの各地点での存在確率は、
        // 移動確率を表す行列と、t-1での存在確率ベクトルの、行列ベクトル積である。
        resultVectors[t] = matmul.MatVec(probMat, resultVectors[t-1])
    }
    return resultVectors
}

// パラメーターcとsを受け取り、バッタの移動確率を表す行列の値を計算する
func getProbMat(c float64, s float64) [N][N]float64 {
    // 問に示された「バッタGの振る舞い」行列を転置したものと、
    // 時刻tでのバッタの存在確率ベクトルとの行列ベクトル積を取ると、
    // その計算結果が時刻t+1でのバッタの存在確率ベクトルとなっている。
    return matmul.Transpose([N][N]float64{
        {s + (1-s)*(1-c), (1 - s) * c, 0, 0, 0},
        {(1 - s) * (1 - c), s, (1 - s) * c, 0, 0, 0},
        {0, (1 - s) * (1 - c), s, (1 - s) * c, 0, 0},
        {0, 0, (1 - s) * c, s, (1 - s) * (1 - c), 0},
        {0, 0, 0, (1 - s) * c, s, (1 - s) * (1 - c)},
        {0, 0, 0, 0, (1 - s) * c, s + (1-s)*(1-c)},
    })
}
```