

**Міністерство освіти та науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет прикладної математики
Кафедра системного програмування і спеціалізованих комп’ютерних
систем**

Лабораторна робота №1

з дисципліни
“Архітектура комп’ютерів. Апаратне забезпечення”

на тему: **«Ознайомлення з програмними засобами мови Go»**

Виконали: Бобров В.С
Воронін М.Г.
Висовень Б.П.

Студенти групи КВ-63

Перевірів(ла): _____

Завдання на роботу:

Реалізація системи лінійних рівнянь методом оберненої матриці.

Код програми:

```
package main
```

```
import "fmt"
```

```
func Matrix_Inicialization(A [][]float32, n int) [][]float32 {  
    fmt.Println("Enter Matrix :")  
    for index := 0; index < n; index++ {  
        tmp := make([]float32, n)  
        for j := 0; j < n; j++ {  
            fmt.Scanf("%f", &tmp[j])  
        }  
        A = append(A, tmp)  
    }  
    return A[:]  
}
```

```
func Matrix_E(E [][]float32, n int) [][]float32 {  
    for index := 0; index < n; index++ {  
        tmp := make([]float32, n)  
        for j := 0; j < n; j++ {  
            tmp[j] = 0  
        }  
        E = append(E, tmp)  
    }  
    return E[:]  
}
```

```
func Vector_Inicialization(b []float32, n int) []float32 {  
    fmt.Println("Enter Vector :")  
    tmp := make([]float32, n)  
    for j := 0; j < n; j++ {  
        fmt.Scanf("%f", &tmp[j])  
    }  
    b = append(tmp)  
    return b[:]  
}
```

```
func Vector_X(n int) []float32 {  
    var X []float32  
    tmp := make([]float32, n)  
    for j := 0; j < n; j++ {  
        tmp[j] = 0  
    }  
}
```

```

    X = append(tmp)
    return X[:]
}

func F1(n int, b []float32, E [][]float32) []float32 {
    var X []float32 = Vector_X(n)
    for i := 0; i < n; i++ {
        var S float32 = 0
        for j := 0; j < n; j++ {
            S = S + E[i][j]*b[j]
            X[i] = S
        }
        fmt.Printf("%0.3f \t", X[i])
    }
    return X[:]
}

func Vector_E(n int, E [][]float32) [][]float32 {
    for i := 0; i < n; i++ {
        for j := 0; j < n; j++ {
            if i == j {
                E[i][j] = 1
            } else {
                E[i][j] = 0
            }
        }
    }
    return E[:]
}

func F2(n int, A [][]float32, E [][]float32, y float32, yes float32) ([][]float32, []
[]float32, float32, float32) {

    for k := 0; k < n; k++ {
        if A[k][k] == 0 {
            y = 0
        } else {
            y = 1 / A[k][k]
        }
        for j := 0; j < n; j++ {
            A[k][j] = A[k][j] * y
            E[k][j] = E[k][j] * y
        }

        for i := k + 1; i < n; i++ {

```

```

        yes = A[i][k]
        for z := 0; z < n; z++ {
            A[i][z] = A[i][z] - A[k][z]*yes
            E[i][z] = E[i][z] - E[k][z]*yes
        }
    }
}
return E[:, A[:, y, yes
}
func F3(n int, A [][]float32, E [][]float32, yes float32) ([][]float32, [][]float32,
float32) {
    for k := n - 1; k >= 0; k-- {
        for i := k - 1; i >= 0; i-- {
            yes = A[i][k]
            for z := n - 1; z >= 0; z-- {
                A[i][z] = A[i][z] - A[k][z]*yes
                E[i][z] = E[i][z] - E[k][z]*yes
            }
        }
    }
    return E[:, A[:, yes
}

```

```

/*****-5 7 9 8 29 -11*****/
func main() {
    var A [][]float32
    var b []float32
    fmt.Println("Please Enter matrix size from 2 to 4")
    var n int
    fmt.Scanf("%d", &n)
    var E [][]float32
    var y float32
    var yes float32
    A = Matrix_Inicialization(A[:, n)
    E = Matrix_E(E[:, n)
    b = Vector_Inicialization(b[:, n)
    E = Vector_E(n, E)
    E, A, y, yes = F2(n, A, E, y, yes)
    E, A, yes = F3(n, A, E, yes)
    F1(n, b, E)
}

```

Код тест файлу

```
package main
```

```
import (  
    "testing"  
)
```

```
type testpair struct {  
    matrix [][]float32  
    vector []float32  
    n      int  
    result []float32  
}
```

```
var tests = []testpair{  
    {[][]float32{{-5, 7}, {9, 8}}, []float32{29, -11}, 2, []float32{-3, 2}},  
    {[][]float32{{1, -1}, {3, -2}}, []float32{3, 1}, 2, []float32{-5, -8}},  
    {[][]float32{{1, 2}, {3, -1}}, []float32{8, 3}, 2, []float32{2, 3}},  
}
```

```
func TestAverage(t *testing.T) {  
    var A [][]float32  
    var b []float32  
    var E [][]float32  
    var y float32  
    var yes float32  
    var i int = 0  
    for _, pair := range tests {  
        A = pair.matrix  
        E = Matrix_E(E[:], pair.n)  
        b = pair.vector  
        E = Vector_E(pair.n, E)  
        E, A, y, yes = F2(pair.n, A, E, y, yes)  
        E, A, yes = F3(pair.n, A, E, yes)  
        var X = F1(pair.n, b, E)  
        if X[i] != pair.result[i] {  
            t.Error(  
                "For", pair.matrix,  
                "expected", pair.result,  
                "got", X,  
            )  
        }  
        i++  
    }  
}
```

Lab1.go - Lab1 - Visual Studio Code

File Edit Selection View Go Debug Terminal Help

EXPLORER

Lab1.go x Lab1_test.go

OPEN EDITORS

LAB1

.vscode

Lab1_test.go

Lab1.go 9+

```
99     A[i][z] = A[i][z] - A[k][z]*yes
100     E[i][z] = E[i][z] - E[k][z]*yes
101     }
102 }
103 }
104 return E[:], A[:], yes
105 }
106
107 /*****5 7 9 8 29 -11*****/
108 func main() {
109     var A [][]float32
110     var b []float32
111     fmt.Println("Please Enter matrix size from 2 to 4")
112     var n int
113     fmt.Scanf("%d", &n)
114     var E [][]float32
115     var y float32
116     var yes float32
117     A = Matrix_Initialization(A[:], n)
118     E = Matrix_E(E[:], n)
119     b = Vector_Initialization(b[:], n)
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL

1: sh

```
exit status 2
FAIL: /home/vlad/Desktop/Lab1 0.006s
$ go run Lab1.go
Please Enter matrix size from 2 to 4
3
Enter Matrix :
3 1 1 5 -1 -1 1 -1 5
Enter Vector :
-2 10 -12
1.000 -2.000 -3.000 $
```

OUTLINE

Ln 112, Col 14 Tab Size: 4 UTF-8 LF Go

Lab1_test.go - Lab1 - Visual Studio Code

File Edit Selection View Go Debug Terminal Help

EXPLORER

Lab1.go x Lab1_test.go x

OPEN EDITORS

LAB1

.vscode

Lab1_test.go

Lab1.go 9+

```
7 type testpair struct {
8     matrix [][]float32
9     vector []float32
10    n int
11    result []float32
12 }
13
14 var tests = []testpair{
15     {[][]float32{{-5, 7}, {9, 8}}, []float32{29, -11}, 2, []float32{-3, 2}},
16     {[][]float32{{1, 2}, {3, -1}}, []float32{8, 3}, 2, []float32{2, 3}},
17 }
18
19 run test | debug test
20 func TestAverage(t *testing.T) {
21     var A [][]float32
22     var b []float32
23     var E [][]float32
24     var y float32
25     var yes float32
26     var i int = 0
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL

1: sh

```
$ go test
-3.000 2.000 2.000 3.000 --- FAIL: TestAverage (0.00s)
    Lab1_test.go:35: For [[1 0] [-0 1]] expected [2 3] got [1.9999995 3.0000002]
FAIL
exit status 1
FAIL: /home/vlad/Desktop/Lab1 0.002s
$
```

OUTLINE

Ln 17, Col 1 Tab Size: 4 UTF-8 LF Go