# Federal Urdu University of Arts, Sciences & Technology, Karachi

**Gulshan-e-Iqbal Campus**


**Name:** **Shoaib Ahmed**

**Father's Name:** **Ahmed Hussain**

**Major Dept:** **Science & Technology**

**Dept:** **Computer Science**

**Class/Section:** **BS-6 / C** **Morning**

**Batch:** **17** **Regular**

**Roll:** **27**


**Course:** **Numerical Computing**

**Subject:** **Assignment 2 (Gaussian Elimination)**

**Instructor:** **Prof. Syed Akhter Raza**

# Assignment 2 (Gaussian Elimination)

Q: Make a complete code of Naive Gaussian Elimination method in R both steps with output of random draw of matrix. Matrix of all students must be unique due to randomly drawn elements. the size of matrix should be at leat 5 X 5. you have to code both steps

1) Forward Elimination

2) Back substitution

submit in a pdf file Algorithm + Code + Output.

Ans: **Algorithm (Forward Elimination) :**

Start
Declare matrix a,b
For k = 1 to n-1
For i = k+1 to n
Factor = $a_{i,k} / a_{k,k}$
For j = k+1 to n
$a_{i,j} = a_{i,j}$ - factor * $a_{k,j}$
End for
$b_i = b_i$ - factor * $b_k$
End for
End for
Stop

## Code (Forward Elimination):

```
a <- array(c(6, 12, 3, -6,10,
             -2,-8,-13,4,2,
             2,6,9,1,3,
             4,10,3,-18,5,
             5,7,8,-16,19), dim=c(5,5))
b <- array(c(16,26,-19,-34,44),  dim=c(5,1))

print(a)
print(b)

for(k in 1:4){
  for (i in seq(k+1,5)) {
    factor1 <- ( a[i,k] / a[k,k]  )
    for (j in seq(k,5)) {
      a[i,j] <- a[i,j] - factor1  * a[k,j]
    }
    b[i] <- b[i] - factor1  *  b[k]
  }
}
print(a)
print(b)
```

## Output (Forward Elimination):

```
> a <- array(c(6, 12, 3, -6,10,
+              -2,-8,-13,4,2,
+              2,6,9,1,3,
+              4,10,3,-18,5,
+              5,7,8,-16,19), dim=c(5,5))
> b <- array(c(16,26,-19,-34,44),  dim=c(5,1))
>
> print(a)
     [,1] [,2] [,3] [,4] [,5]
[1,]    6   -2    2    4    5
[2,]   12   -8    6   10    7
[3,]    3  -13    9    3    8
[4,]   -6    4    1  -18  -16
[5,]   10    2    3    5   19
> print(b)
     [,1]
[1,]   16
[2,]   26
[3,]  -19
[4,]  -34
[5,]   44
>
> for(k in 1:4){
+   for (i in seq(k+1,5)) {
+     factor1 <- ( a[i,k] / a[k,k]  )
+     for (j in seq(k,5)) {
+       a[i,j] <- a[i,j] - factor1  * a[k,j]
+     }
```

```
+     for (j in seq(k,5)) {
+       a[i,j] <- a[i,j] - factor1  * a[k,j]
+     }
+     b[i] <- b[i] - factor1  *  b[k]
+   }
+ }
> print(a)
     [,1] [,2] [,3] [,4]      [,5]
[1,]    6   -2    2    4    5.0000
[2,]    0   -4    2    2   -3.0000
[3,]    0    0    2   -5   14.5000
[4,]    0    0    0   -3  -41.5000
[5,]    0    0    0    0 -104.7778
> print(b)
     [,1]
[1,]   16
[2,]   -6
[3,]   -9
[4,]   -3
[5,]   13
>
```

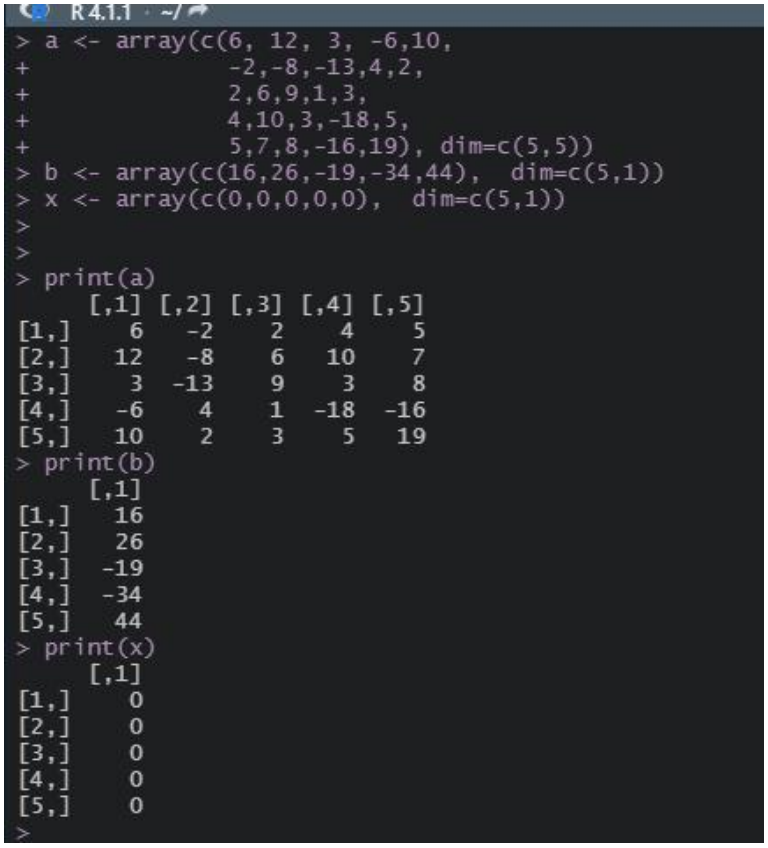
## Algorithm (Backward Substitution) :

Start
Declare matrix c
$x_n = b_n / a_{n,n}$
For i = n-1 down to 1
Sum = $b_i$
for j = i+1 to n
Sum = sum - $a_{i,j} * x_j$
End for
$x_i$ = sum / $a_{i,i}$
End for
Stop

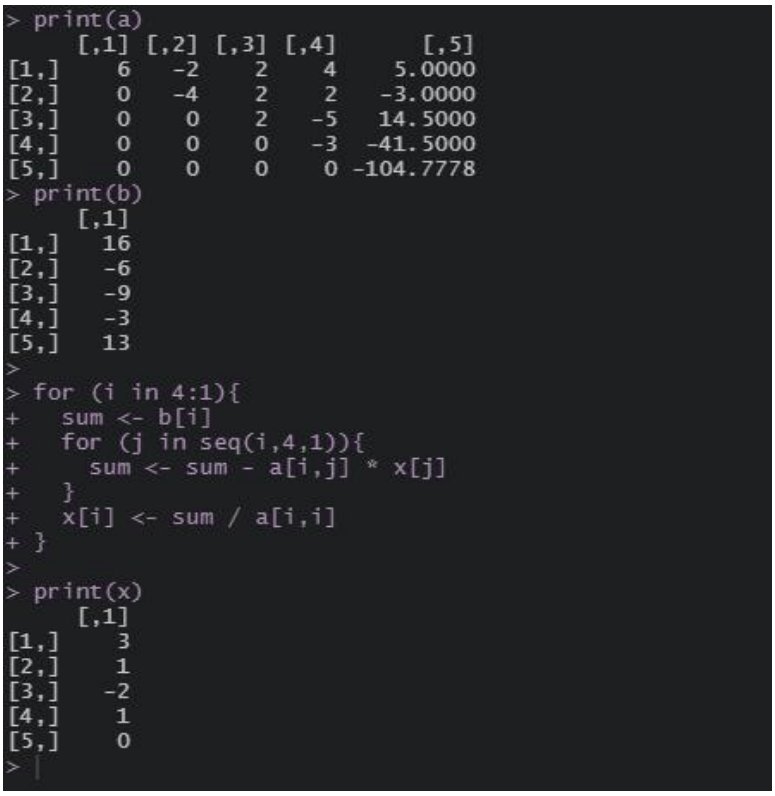
## Code (Backward Substitution):

```
Source on Save
a <- array(c(6, 12, 3, -6,10,
             -2,-8,-13,4,2,
             2,6,9,1,3,
             4,10,3,-18,5,
             5,7,8,-16,19), dim=c(5,5))
b <- array(c(16,26,-19,-34,44),  dim=c(5,1))
x <- array(c(0,0,0,0,0),  dim=c(5,1))


print(a)
print(b)
print(x)

for(k in 1:4){
  for (i in seq(k+1,5)) {
    factor1 <- ( a[i,k] / a[k,k]  )
    for (j in seq(k,5)) {
      a[i,j] <- a[i,j] - factor1  * a[k,j]
    }
    b[i] <- b[i] - factor1  *  b[k]
  }
}
print(a)
print(b)

for (i in 4:1){
  sum <- b[i]
  for (j in seq(i,4,1)){
    sum <- sum - a[i,j] * x[j]
  }
  x[i] <- sum / a[i,i]
}

print(x)
```

# Output (Backward Substitution):



```
R 4.1.1 · ~/
> a <- array(c(6, 12, 3, -6,10,
+              -2,-8,-13,4,2,
+              2,6,9,1,3,
+              4,10,3,-18,5,
+              5,7,8,-16,19), dim=c(5,5))
> b <- array(c(16,26,-19,-34,44),  dim=c(5,1))
> x <- array(c(0,0,0,0,0),  dim=c(5,1))
>
>
> print(a)
     [,1] [,2] [,3] [,4] [,5]
[1,]    6   -2    2    4    5
[2,]   12   -8    6   10    7
[3,]    3  -13    9    3    8
[4,]   -6    4    1  -18  -16
[5,]   10    2    3    5   19
> print(b)
     [,1]
[1,]   16
[2,]   26
[3,]  -19
[4,]  -34
[5,]   44
> print(x)
     [,1]
[1,]    0
[2,]    0
[3,]    0
[4,]    0
[5,]    0
>
```



```
> print(a)
     [,1] [,2] [,3] [,4]      [,5]
[1,]    6   -2    2    4    5.0000
[2,]    0   -4    2    2   -3.0000
[3,]    0    0    2   -5   14.5000
[4,]    0    0    0   -3  -41.5000
[5,]    0    0    0    0 -104.7778
> print(b)
     [,1]
[1,]   16
[2,]   -6
[3,]   -9
[4,]   -3
[5,]   13
>
> for (i in 4:1){
+   sum <- b[i]
+   for (j in seq(i,4,1)){
+     sum <- sum - a[i,j] * x[j]
+   }
+   x[i] <- sum / a[i,i]
+ }
>
> print(x)
     [,1]
[1,]    3
[2,]    1
[3,]   -2
[4,]    1
[5,]    0
>
```