SIKSHA 'O' ANUSANDHAN DEEMED UNIVERSITY

MATRIX MANIPULATOR

PRESENTED BY:

PRERNA BHARTI – 1641012048

NIKHIL RANJAN NAYAK - 1641012040

SATYASMITH RAY – 1641012108

BAIBHAV SWAIN - 1641012206

SONU BINAY – 1641012316

(CSE ‘F’ 4th SEMESTER)

Problem Statement

To develop matrix operation calculator using MIPS.

It consists of basic operation like addition, subtraction, scaling and transpose and multiplication.

- Addition or subtraction is accomplished by adding or subtracting corresponding elements.
- Scaling is used to scale up/down the matrix with a constant using scalar multiplication.
- Transpose of matrix is that in which all the rows of a given matrix is transformed into columns and vice-versa.
- Multiplication of matrices is also done using MIPS code by rows of first matrix with the columns of second matrix.
- The determinant of a matrix is a scalar property of that matrix, which can be thought of physically as the volume enclosed by the row vectors of the matrix. Only square matrices have determinants.

Brief Description

Manipulator system will be required to carry out various operations and provide meaning to these combined structures.

- In Matrix Addition and subtraction, we assume the two matrices given in the data segment. All we have to do is to start with the base address, add the respective elements of the two matrices, increment the address by four and do the same till the length is reached.
- While approaching for the Scaling of matrices there are many prospects to show scaling. We chose the one with scalar multiplication of the matrix with a constant.
- The next operation assigned to us is transpose in which we just have to reverse the rows with the columns and the columns with the rows simultaneously.
- To find determinant we use the formula A[0][0] x ((A[1][1]xA[2][2])-(A[2][1]xA[1][2]))-A[0][1]x((A[1][0]xA[2][2])-(A[2][0]xA[1][2]))+A[0][2]x((A[1][0]xA[2][1])-(A[2][0]xA[1][1])) as we have considered a 3x3 squared matrix.
- The last operation is multiplication of the matrix which is done multiplying the rows of first matrix with the columns of second matrix.
- If the user enters invalid option then error message is displayed.

Algorithm

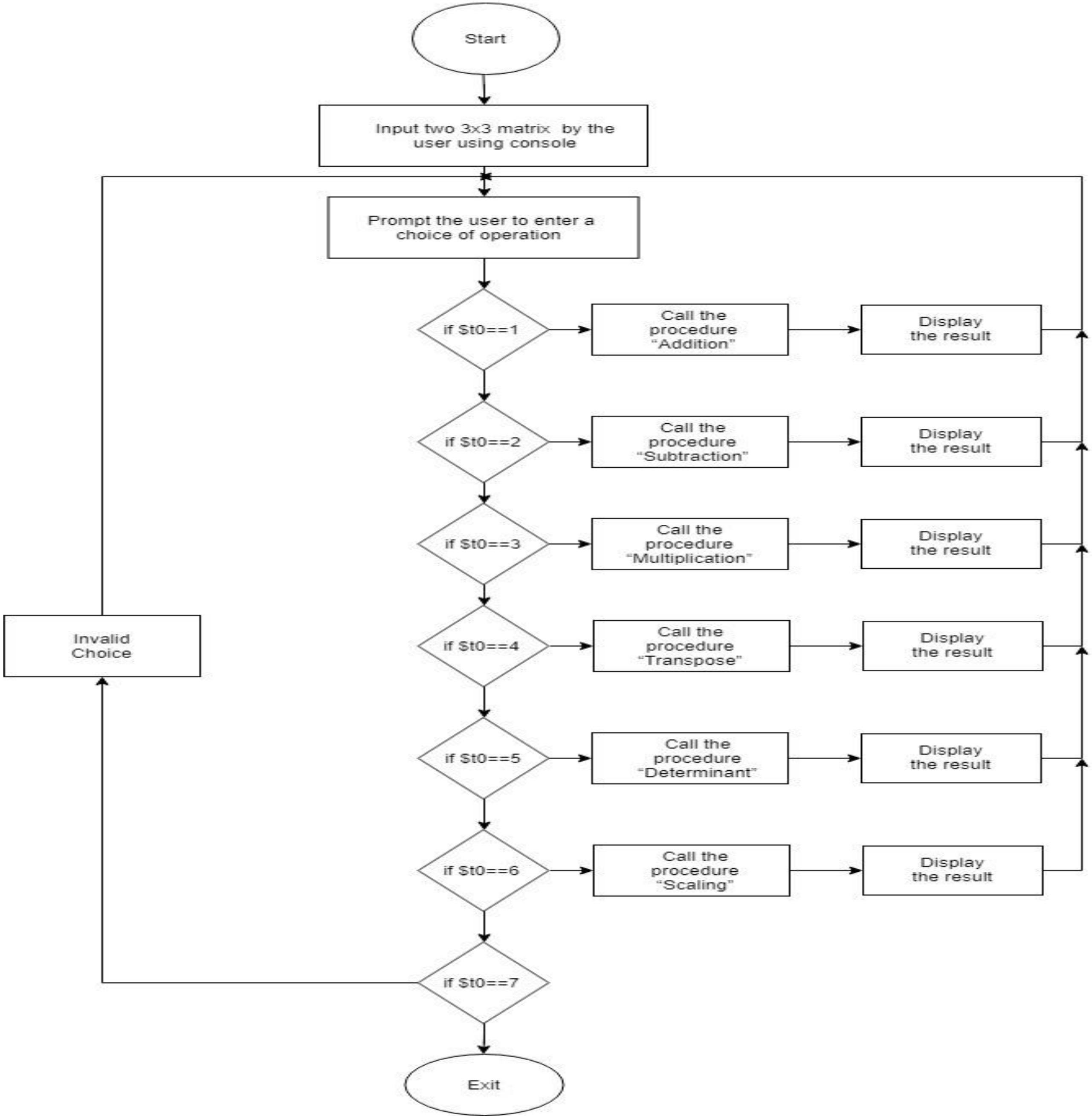
Step 1. Input the two matrixes each of size 3x3 from the user in the console using various syscall services.

Step 2. Prompt the user to enter the choice of the operation and store it in $t0 register.

Step 3.

I. If the content of $t0 register=1 then perform addition of the two matrices (using addition function).
II. Else If the content of $t0 register=2 then perform subtraction of the two matrices (using subtraction function).
III. Else If the content of $t0 register=3 then perform multiplication of two matrices (using multiplication function).
IV. Else If the content of $t0 register=4 then perform transpose of first matrix (using transpose function).
V. Else If the content of $t0 register= 5 then find determinant of first matrix (using determinant function).
VI. Else If the content of $t0 register= 6 then perform scaling for first matrix by taking a scaling element(using scaling function).
VII. Else If the content of $t0 register= 7 then exit.
VIII. Else For any other choice, display the error message.

Flow Chart



Start

Input two 3x3 matrix by the user using console

Prompt the user to enter a choice of operation

if $t0==1

Call the procedure "Addition"

Display the result

if $t0==2

Call the procedure "Subtraction"

Display the result

if $t0==3

Call the procedure "Multiplication"

Display the result

if $t0==4

Call the procedure "Transpose"

Display the result

if $t0==5

Call the procedure "Determinant"

Display the result

if $t0==6

Call the procedure "Scaling"

Display the result

if $t0==7

Invalid Choice

Exit

C Implementation

```
#include<stdio.h>
void addition(int A[][3],int B[][3],int n);
void subtraction(int A[][3],int B[][3],int n);
void multiplication(int A[][3],int B[][3],int n);
void transpose(int A[][3],int n);
void determinant(int A[][3]);
void scaling(int A[][3],int n);
void print_matrix(int A[][3],int n);
int main()
{
        int n,i,j,choice;
        printf("Enter dimension of array=> ");
        scanf("%d",&n);
        int A[3][3],B[3][3];
        printf("Enter elements in first array\n");
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
```

```
    for(j=0;j<n;j++){
            scanf("%d",&A[i][j]);
        }
    }
    printf("Enter elements in second
array\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            scanf("%d",&B[i][j]);
        }
    }
    printf("\nFirst entered matrix is as
follows:\n");
    print_matrix(A,n);
    printf("\nSecond entered matrix is as
follows:\n");
    print_matrix(B,n);
    printf("\nMENU\n");
    printf("1.ADDITION\n");
    printf("2.SUBTRACTION\n");
    printf("3.MULTIPLICATION\n");
    printf("4.TRANSPOSE\n");
    printf("5.DETERMINANT\n");
    printf("6.SCALING\n");
printf("7.EXIT\n");
printf("Enter your choice=> ");
scanf("%d",&choice);
while(choice!=7){
        switch(choice){
                case 1:
                        addition(A,B,n);
                        break;
                case 2:
                        subtraction(A,B,n);
                        break;
                case 3:
                        multiplication(A,B,n);
                        break;
                case 4:
                        transpose(A,n);
                        break;
                case 5:
                        determinant(A);
                        break;
```

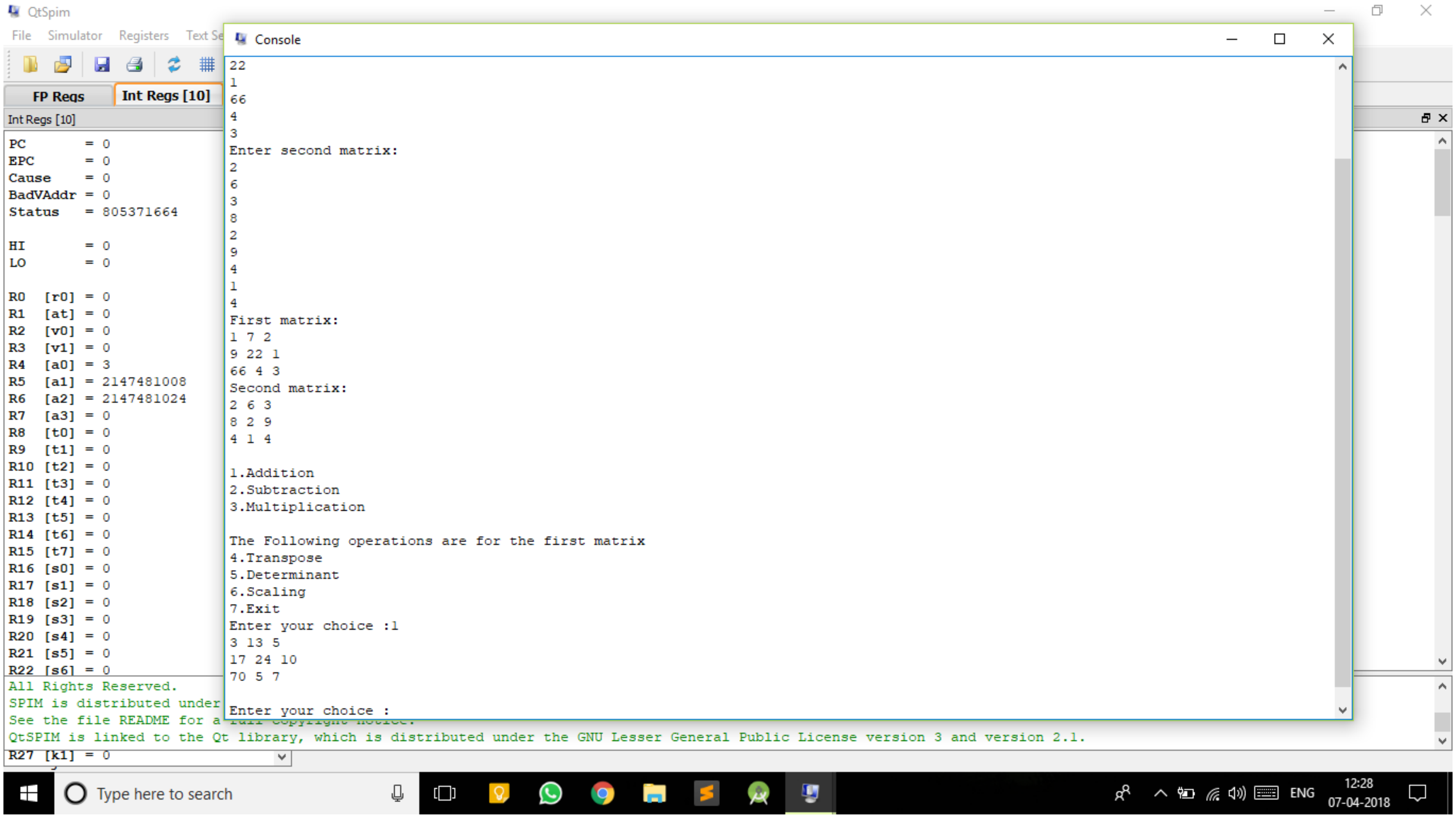
```
             case 6:
                scaling(A,n);
                break;
            default:
                printf("\nINVALID CHOICE!");
        }
        printf("\nEnter your choice=> ");
        scanf("%d",&choice);
    }
    printf("\nEXITING");
    return 0;
}
void addition(int A[][3],int B[][3],int n)
{
    int i,j,sum=0;
    printf("\nResultant matrix:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            sum=A[i][j]+B[i][j];
            printf("%d ",sum);
        }
        printf("\n");
    }
}
```

```c
void subtraction(int A[][3],int B[][3],int n)
{
        printf("\nResultant matrix:\n");
        int i,j,diff=0;
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
                        diff=A[i][j]-B[i][j];
                        printf("%d ",diff);
                }
                printf("\n");
        }
}
void multiplication(int A[][3],int B[][3],int n)
{
        int i,j,k,sum;
        printf("\nResultant matrix:\n");
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
                        sum=0;
                        for(k=0;k<n;k++){
                                sum+=A[i][k]*B[k][j];
                        }
                        printf("%d ",sum);
                }
}
        printf("\n");
    }
}
void transpose(int A[][3],int n)
{
    int i,j,t;
    printf("\nResultant matrix:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            t=A[j][i];
            printf("%d ",t);
        }
        printf("\n");
    }
}
void determinant(int A[][3])
{
    int d=A[0][0]*((A[1][1]*A[2][2])-(A[2][1]*A[1][2]))-A[0][1]*((A[1][0]*A[2][2])-
(A[2][0]*A[1][2]))+A[0][2]*((A[1][0]*A[2][1])-(A[2][0]*A[1][1]));
    printf("\nDeterminant result= %d",d);
}
void scaling(int A[][3],int n)
```

```
{
        int scaling_factor,scale,i,j;
        printf("Enter a number for performing scaling=> ");
        scanf("%d",&scaling_factor);
        printf("\nResultant matrix:\n");
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
                        scale=A[i][j]*scaling_factor;
                        printf("%d ",scale);
                }
                printf("\n");
        }

}
void print_matrix(int A[][3],int n)
{
        int i,j;
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
                        printf("%d ",A[i][j]);
                }
                printf("\n");
        }
}
```

Screenshots

ADDITION



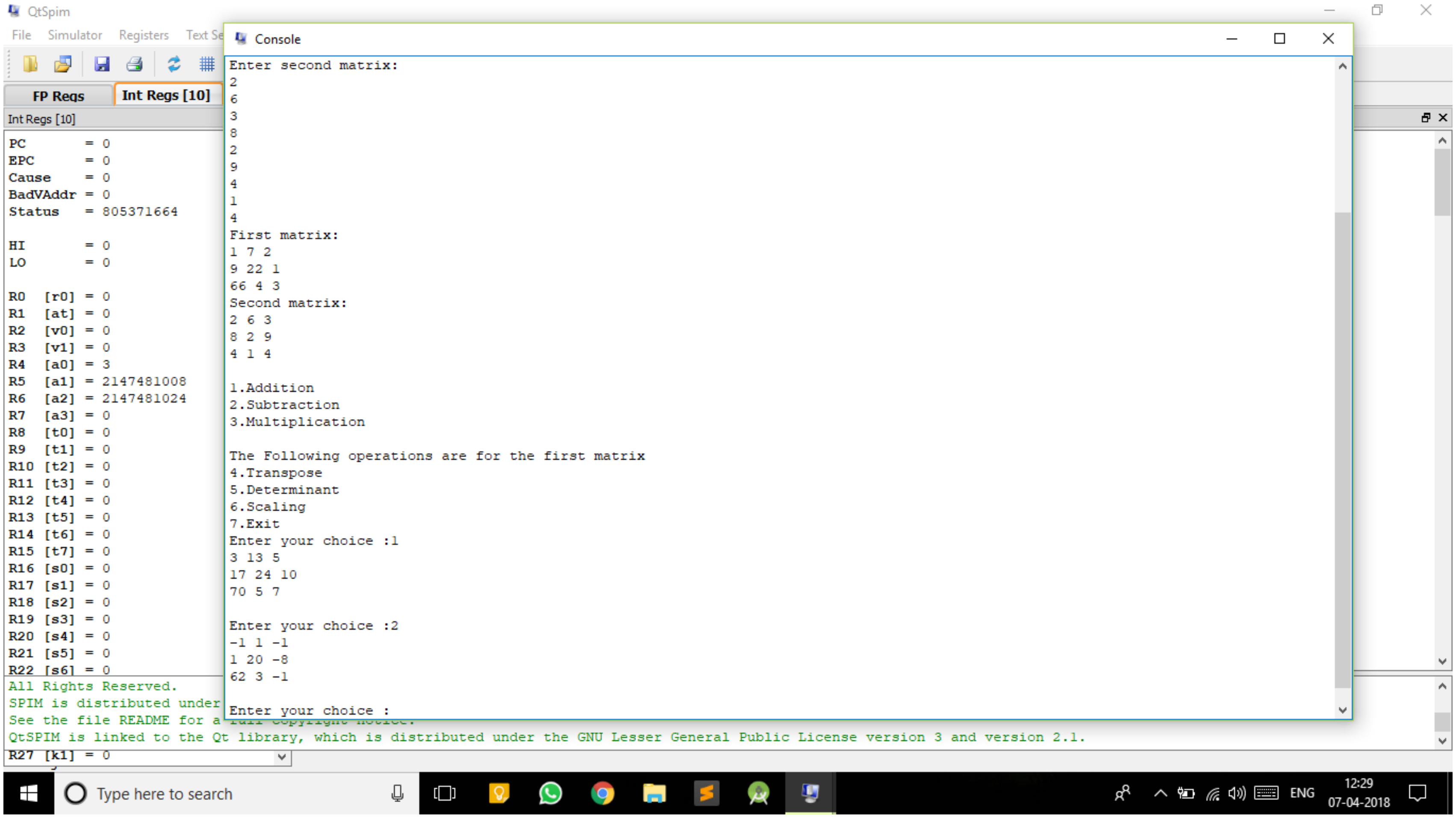
```
PC       = 0
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 805371664

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 0
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 2147481008
R6  [a2] = 2147481024
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
```

```
22
1
66
4
3
Enter second matrix:
2
6
3
8
2
9
4
1
4
First matrix:
1 7 2
9 22 1
66 4 3
Second matrix:
2 6 3
8 2 9
4 1 4

1.Addition
2.Subtraction
3.Multiplication

The Following operations are for the first matrix
4.Transpose
5.Determinant
6.Scaling
7.Exit
Enter your choice :1
3 13 5
17 24 10
70 5 7

Enter your choice :
```

```
All Rights Reserved.
SPIM is distributed under
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

```
R27 [k1] = 0
```

SUBTRACTION

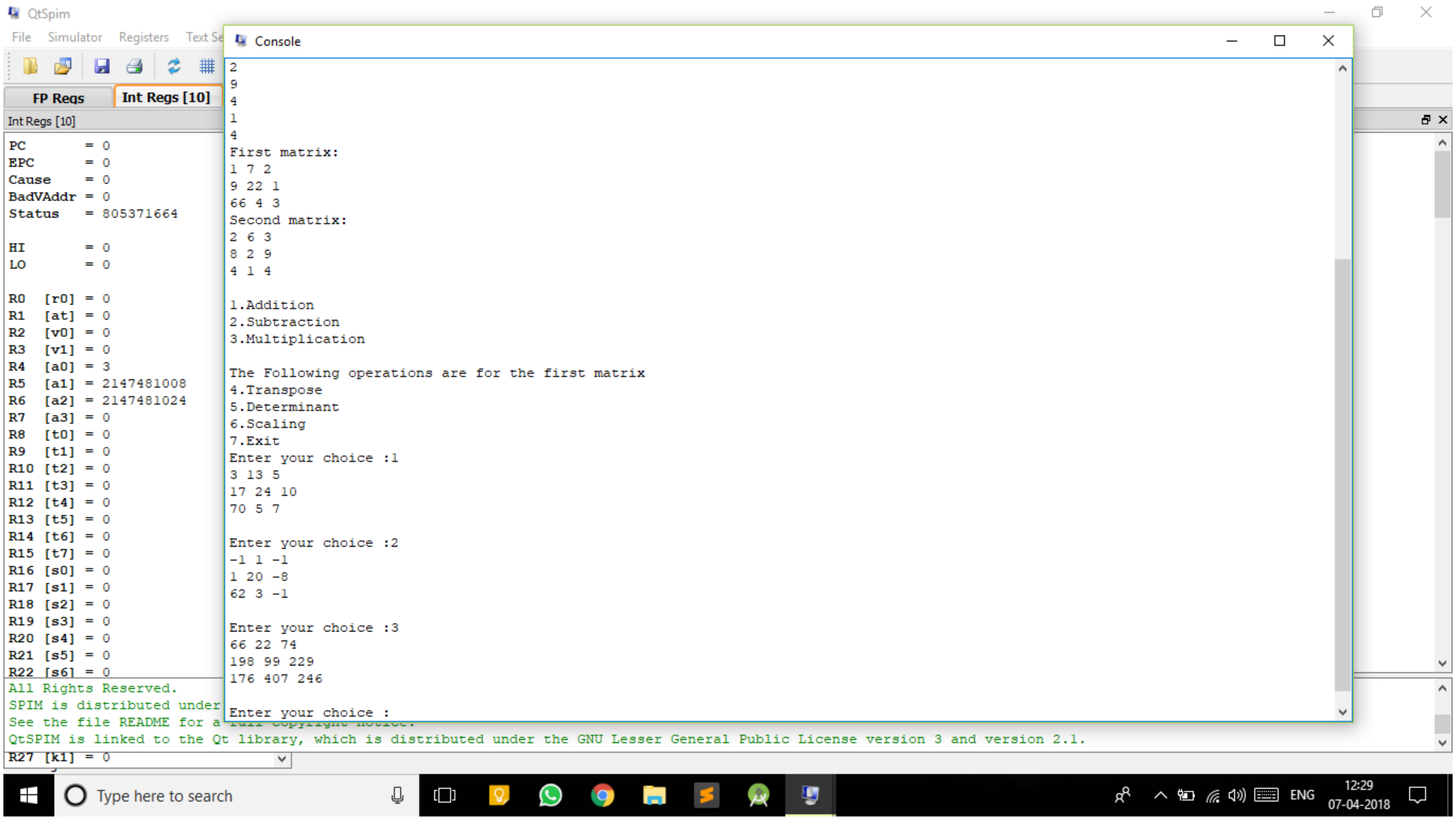


```
PC       = 0
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 805371664

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 0
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 2147481008
R6  [a2] = 2147481024
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
```

```
Enter second matrix:
2
6
3
8
2
9
4
1
4
First matrix:
1 7 2
9 22 1
66 4 3
Second matrix:
2 6 3
8 2 9
4 1 4

1.Addition
2.Subtraction
3.Multiplication

The Following operations are for the first matrix
4.Transpose
5.Determinant
6.Scaling
7.Exit
Enter your choice :1
3 13 5
17 24 10
70 5 7

Enter your choice :2
-1 1 -1
1 20 -8
62 3 -1

Enter your choice :
```

```
All Rights Reserved.
SPIM is distributed under
See the file README for a
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
R27 [k1] = 0
```

MULTIPLICATION



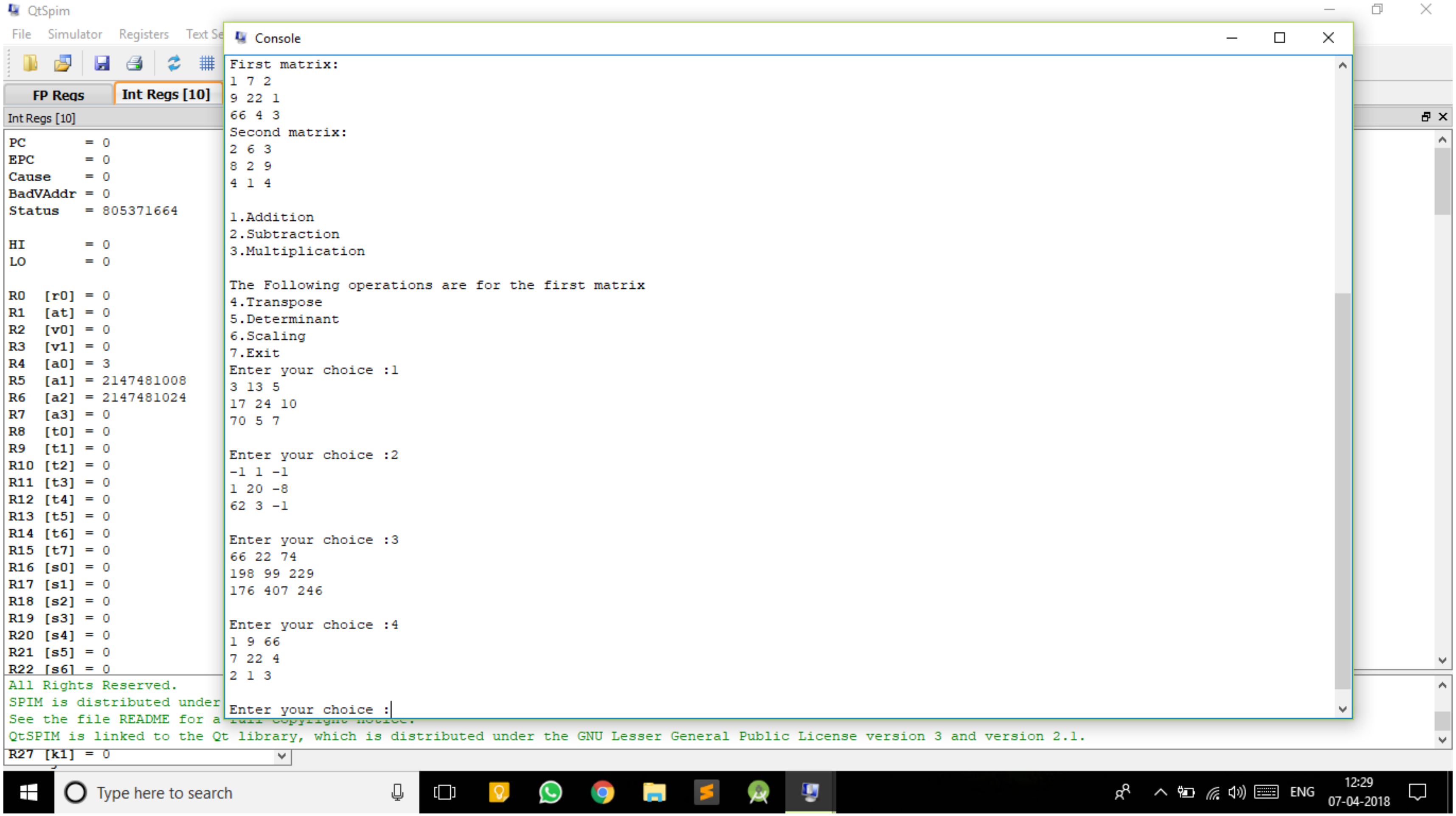
```
PC       = 0
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 805371664

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 0
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 2147481008
R6  [a2] = 2147481024
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
```

```
2
9
4
1
4
First matrix:
1 7 2
9 22 1
66 4 3
Second matrix:
2 6 3
8 2 9
4 1 4

1.Addition
2.Subtraction
3.Multiplication

The Following operations are for the first matrix
4.Transpose
5.Determinant
6.Scaling
7.Exit
Enter your choice :1
3 13 5
17 24 10
70 5 7

Enter your choice :2
-1 1 -1
1 20 -8
62 3 -1

Enter your choice :3
66 22 74
198 99 229
176 407 246

Enter your choice :
```

```
All Rights Reserved.
SPIM is distributed under
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

```
R27 [k1] = 0
```

TRANSPOSE

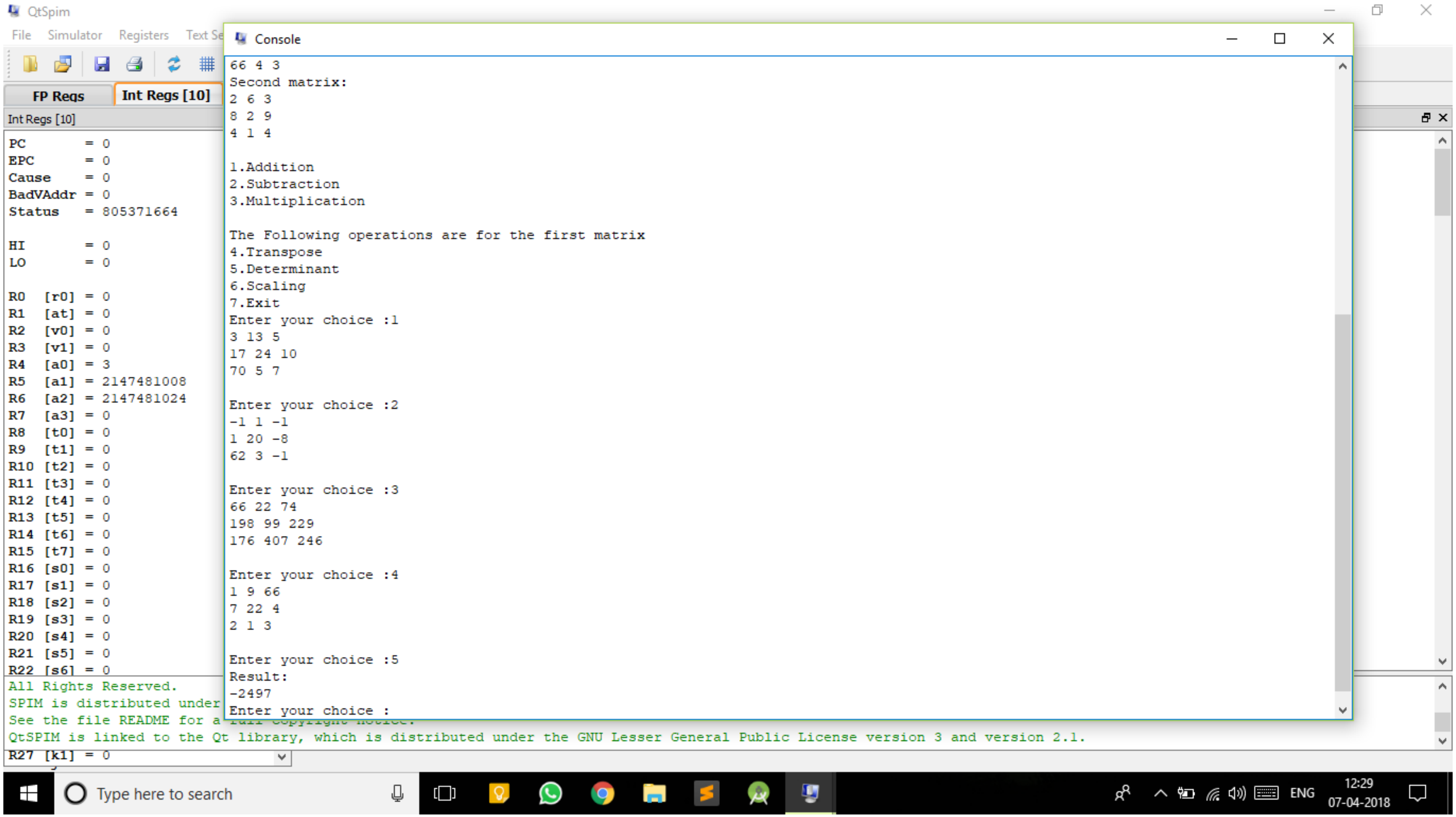


```
First matrix:
1 7 2
9 22 1
66 4 3
Second matrix:
2 6 3
8 2 9
4 1 4

1.Addition
2.Subtraction
3.Multiplication

The Following operations are for the first matrix
4.Transpose
5.Determinant
6.Scaling
7.Exit
Enter your choice :1
3 13 5
17 24 10
70 5 7

Enter your choice :2
-1 1 -1
1 20 -8
62 3 -1

Enter your choice :3
66 22 74
198 99 229
176 407 246

Enter your choice :4
1 9 66
7 22 4
2 1 3

Enter your choice :
```

```
PC       = 0
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 805371664

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 0
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 2147481008
R6  [a2] = 2147481024
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R27 [k1] = 0
```

```
All Rights Reserved.
SPIM is distributed under
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

DETERMINANT

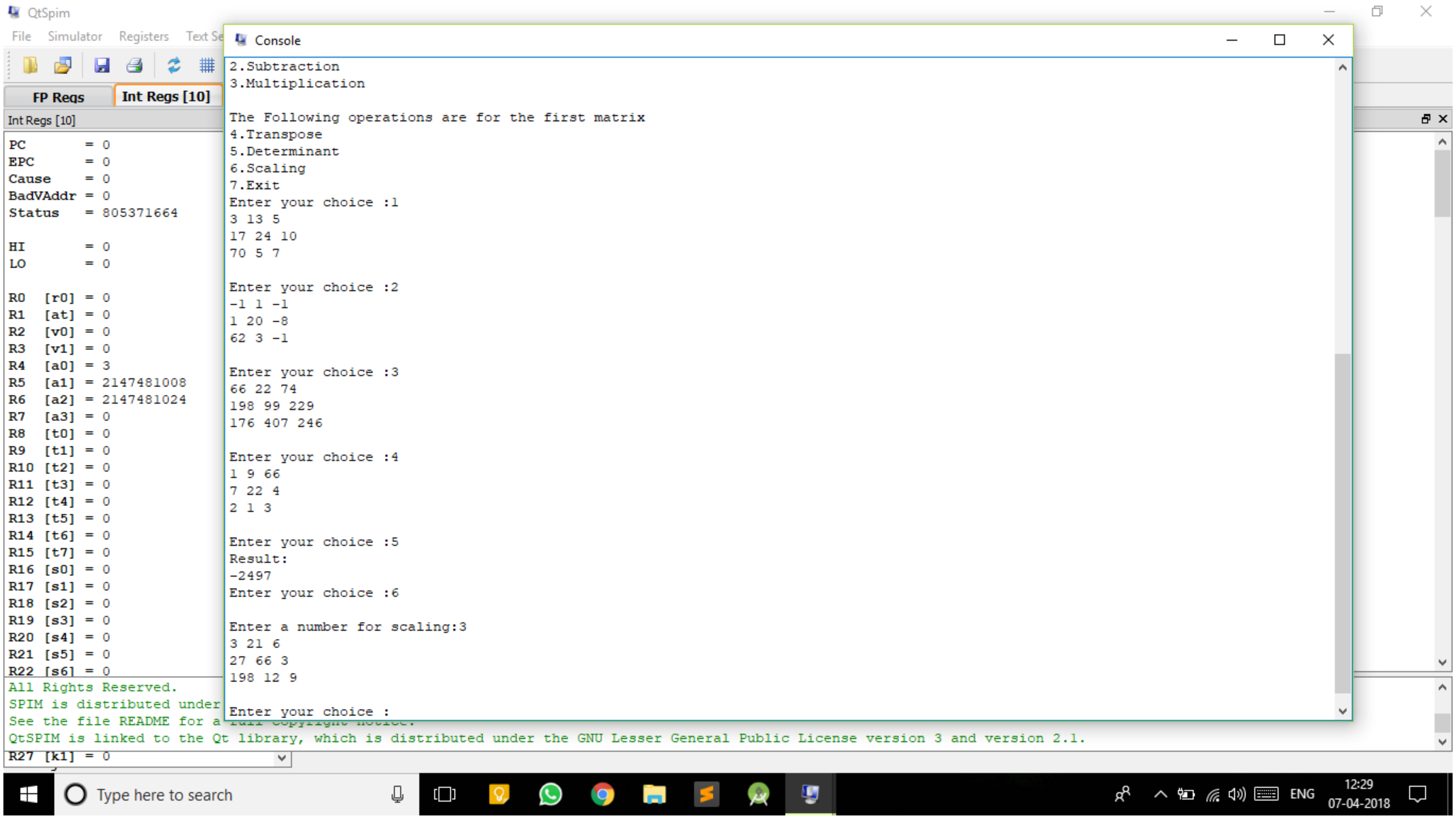


```
PC       = 0
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 805371664

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 0
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 2147481008
R6  [a2] = 2147481024
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
```

```
66 4 3
Second matrix:
2 6 3
8 2 9
4 1 4

1.Addition
2.Subtraction
3.Multiplication

The Following operations are for the first matrix
4.Transpose
5.Determinant
6.Scaling
7.Exit
Enter your choice :1
3 13 5
17 24 10
70 5 7

Enter your choice :2
-1 1 -1
1 20 -8
62 3 -1

Enter your choice :3
66 22 74
198 99 229
176 407 246

Enter your choice :4
1 9 66
7 22 4
2 1 3

Enter your choice :5
Result:
-2497
Enter your choice :
```

```
All Rights Reserved.
SPIM is distributed under
See the file README for a
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
R27 [k1] = 0
```

SCALING



```
Int Regs [10]
PC       = 0
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 805371664

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 0
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 2147481008
R6  [a2] = 2147481024
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R27 [k1] = 0
```

```
2.Subtraction
3.Multiplication

The Following operations are for the first matrix
4.Transpose
5.Determinant
6.Scaling
7.Exit
Enter your choice :1
3 13 5
17 24 10
70 5 7

Enter your choice :2
-1 1 -1
1 20 -8
62 3 -1

Enter your choice :3
66 22 74
198 99 229
176 407 246

Enter your choice :4
1 9 66
7 22 4
2 1 3

Enter your choice :5
Result:
-2497
Enter your choice :6

Enter a number for scaling:3
3 21 6
27 66 3
198 12 9

Enter your choice :
```

```
All Rights Reserved.
SPIM is distributed under
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

Conclusion

From this project, we got an idea on how to perform different arithmetic operations on matrix like addition, subtraction, multiplication, transpose, determinant and scaling using MIPS code.

References

- Computer Organization and Design
    - By David A Patterson and John L Hennessy.
- New Jersey Institute of Technology eLab
    - http://ecelabs.njit.edu/ece459/lab1.php
- University of Pittsburgh e-Library
    - http://people.cs.pitt.edu/~xujie/cs447/AccessingArray
- Draw.io
    - Online diagram software for making flowcharts