IFoS-2017 → Paper II

5) (b) Write a BASIC program to compute the multiplicative inverse of a non-singular square matrix.

⇒
```
# include <stdio.h>
# include <conio.h>
# include <math.h>
void main()
{
float a[10][10], b[10][10], x;
int i, j, k, n;
Printf("\n Enter the order of the matrix: ");
scanf("%d", &n);
Printf("\n Enter a matrix Row-wise \n");
for(i=1; i<=n; i++)
{
for(j=1; j<= n; j++)
 scanf("%f", &a[i][j]);
 Printf("\n");
}
```

```
for (i=1; i<=n; i++)
{
for(j=1; j<=n; j++)
b[i][j] = 0.0;
b[i][i] = 1.0;
}
for (k=1; k<=n; k++)
for(i=1; i<=n; i++)
{
if (i==k)
continue;
r = a[i][k] / a[k][k];
for (j=1; j<=n; j++)
{
a[i][j] = a[i][j] - r * a[k][j];
b[i][j] = b[i][j] - r * b[k][j];
}
}
for (i=1; i<=n; i++)
for(j=1; j<=n; j++)
b[i][j] = b[i][j] / a[i][i];
printf ("\n The Inverse Matrix is \n");
for(i=1; i<=n; i++)
for(j=1; j<=n; j++)
Printf ("%.2.5f", b[i][j]);
Printf("\n");
}
getch();
}
```

7) (d) Assuming a 32 bit computer representation of signed integers using 2's Complement representation and the two numbers -1 and -1024 and give the answer in 2's Complement representation.

=> in 32 bit Computer =>

1 ≡ 00000000 00000000 00000000 00000001

1's Complement of 1 is,

11111111 11111111 11111111 11111110

So, the 2's complement is

$(-1)_2$ = 11111111 11111111 11111111 11111111

Now,

1024 ≡ 00000000 00000000 00000100 00000000

1's Complement of 1024 is,

11111111 11111111 11111011 11111111

So, the 2's Complement is,

$(-1024)_2$ = 11111111 11111111 11111100 00000000

Now,

```
  11111111 11111111 11111111 11111111
+ 11111111 11111111 11111100 00000000
-----------------------------------------
1 11111111 11111111 11111011 11111111
```

This mean it's (-)ve

For checking, 1's Complement of 11111111 11111111 11111011 11111111

is, 00000000 00000000 00000100 00000000

& 2's Complement is, 00000000 00000000 00000100 00000001

$= 1 \times 2^{10} + 1 \times 2^0$

$= 1024 + 1$

$= 1025$

$= -(-1025)_2$