IFoS-2015 → Paper II

5)(a) store the value of (-1) in hexadecimal in a 32-bit computer.

⇒ 32 bit representation of 1 is
00000000 00000000 00000000 00000001

1's complement of 1 is, 11111111 11111111 11111111 11111110

∴ 2's complement of 1 is,

(-1) ≡ 11111111 11111111 11111111 11111111

Now, we represent it, in hexadecimal format.

So, 1111 1111 1111 1111 1111 1111 1111 1111
F F F F F F F F

So, $(-1)_{10} = (FFFFFFFF)_{16}$

Hence ∴ The value of (-1) is store in hexadecimal format in a 32-bit Computer can be represent as, FFFFFFFF.

8)(a) write a BASIC program to compute the product of two matrices.

⇒
```
# include <stdio.h>
# include <conio.h>
# include <math.h>
void main()
{
int a[10][10], b[10][10], c[10][10];
int i, j, k, m, n, o, p;
clrscr();
printf("\n Enter the number of rows and columns of
                                first matrix: ");
scanf("%d%d", &m, &n);
for(i=0; i<m; i++)
for(j=0; j<n; j++)
scanf("%d", &a[i][j]);
printf("\n Enter the number of rows and columns of
                                second matrix: ");
scanf("%d%d", &o, &p);
```

```
for(i=0 ; i<0; i++)
for (j=0; j<p; j++)
scanf ("%d", &b[i][j]);
if (n==0)
{
 for (i=0 ; i<m; i++)
 for (j=0 ; j<p; j++)
  {
  c[i][j] =0 ;
  for(k=0; k<n; k++)
  c[i][j]=c[i][j] + (a[i][k] * b[k][j]);
  }
}
 else
{
printf ("\n Matrices not Conformable for
                        multiplication");
exit(0);
}
printf("\n The resultant matrix is: \n");
for(i=0 ; i<m, i++)
{
  for(j =0; j<p ; j++)
  printf ("%4d", c[i][j]);
  printf ("\n");
}
getch();
}
```