IFoS-2013 → paper II

6) (b) Convert $(0.231)_5$, $(104.231)_5$ and $(247)_7$ to base 10

⇒ $(0.231)_5 = 2\times5^{-1} + 3\times5^{-2} + 1 + 5^{-3}$

$= \frac{2}{5} + \frac{3}{25} + \frac{1}{125}$

$= \frac{0+15+1}{125} = \frac{66}{125}$

$= (0.528)_{10}$

$$(104.231)_5 = (1\times5^2)+(0\times5^1)+(4\times5^0)+(2\times5^{-1})+(3\times5^{-2})+(1\times5^{-3})$$

$$= 25+0+4+\frac{2}{5}+\frac{3}{25}+\frac{1}{125}$$

$$= 29+\frac{2}{5}+\frac{3}{25}+\frac{1}{125}$$

$$= \frac{3625+50+15+1}{125} = \frac{3691}{125}$$

$$= (29.528)_{10}$$

$$(247)_7 = (2\times7^2)+(4\times7^1)+(4\times7^0)$$

$$= 2\times49+28+4$$

$$= (130)_{10}$$

7) (b) write a algorithm to find the inverse of a given non-singular diagonally dominant square matrix using Gauss-jordan method.

⇒
```
# 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][j] = 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");
  }
  }
```

8) (c) Draw a flow chart for testing whether a given real number is a prime or not.

⇒

```
#include<math.h>
#include<stdio.h>
void main()
{
int n,i,r;
printf("\n Enter the positive integer value N\n");
scanf("%d",&n);
i=2
Step1: if(i<=sqrt(n))
 {
 r = n%i;
 if(r==0)
 {
 printf("%d is not a prime number", n);
 goto end;
 }
else
 {
 i++
 goto sep1;
 }
 }
 printf("%d is a prime Number", n);
 end: printf(" ");
}
```