IFoS -2019 → Paper II

8) (b) state the Newton-Raphson iteration formula to compute a root of an equation $f(x)=0$ and hence write a program in BASIC to compute a root of the equation,

$$\cos x - xe^x = 0$$

lying between 0 and 1. Use DEF function to define $f(x)$ and $f'(x)$.

⇒ By the Newton-Raphson iteration formula, to find isolated roots of an equation $f(x)=0$, we get the $(n+1)$th correct root is,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

program ⇒

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int user_power, i=0, cnt=0, flag=0;
int coef[10]={0};
float x1=0, x2=0, t=0;
float fx1=0, fdx1=0;
void main()
{
clrscr();
printf("\n Enter the total no. of Power: ");
scanf("%d", &user_Power);
for(i=0; i<=user_Power; i++)
{
printf("\n x^%d: ", i);
scanf("%d", &coef[i]);
}
printf("\n");
printf("\n The polynomial is: ");
for(i=user_power; i>=0; i--) // printing coeff
{
printf("%dx^%d", coef[i], i);
}
```

```
printf ("\n initial x = ");
scanf ("%f", &x1);
printf ("\n Iteration  x1  fx1  f'x1 ");
do
{
cnt++;
fx1 = fdx1 = 0;
for (i = user_power; i >= 1; i--)
{
fx1 += coef[i] * (pow(x1, i));
}
fx1 += coef[0];
for (i = user_power; i >= 0; i--)
{
fdx1 += coef[i] * (i * pow(x1, (i-1)));
}
t = x2;
x2 = (x1 - (fx1/fdx1));
x1 = x2;
printf ("\n %d %0.3f %0.3f %0.3f ", cnt, x2, fx1, fdx1);
}
while ((fabs (t-x1)) >= 0.0001);
printf ("\n The root of the equation is %f", x2);
getch();
}
```