Some programs in C - Part 1

Preppared by ratulchakraborty@gmail.com

(1) Integration of the function e^x by Simpsons 1/3rd rule

The C Program

```
#include
#include

float f(float);
float integration(float, float, int);

int main()
{
    int n;
    float lower, upper;

    printf("INPUT LOWER LIMIT, UPPER LIMIT & NUMBER OF SUB-INTERVAL : ");
    scanf("%f%f%d", &lower, &upper, &n);

    printf("\nBY SIMPSON'S 1/3 RULE RESULT = %f ",integration(lower,upper,n));

    return 0;
}

float f(float x)
{
    return exp(-x*x/2);
}

float integration(float a, float b, int n)
{
    int i;
    float h, result = 0;

    h = (b - a)/n;
    for(i=1; i<n; i++){
        result += (i%2 == 0)? 2*f(a + i*h) : 4*f(a + i*h);
    }
    result = h/3 * (f(a) + f(b) + result);

    return result;
}
```

Using this program integrate the following functions w.r.t x:

a. $sin(x)$
b. $sin(x) + cos(x^2)$
c. $log(x)\ sin(x)$
d. $2x\ sin(x) / cos^3(x)$
e. $(sin(x) - cos(x)) / (sin(x) + cos(x))$

(2) Solution of the equation 2x - log(x) - 7 = 0 by Iteration method

The C Program

```
#include
#include

#define MAX_ITER 100

double phi(double);

int main()
{
    int n, count = 0;
    double x, newx, dif, correct;

    printf("FIRST APPROXIMATE ROOT : ");
    scanf("%lf",&x);
    printf("\nCORRECTED TO NUMBER OF DECIMAL PLACES : ");
    scanf("%d",&n);

    correct = pow(10,-(n+1));
    do{
        newx = phi(x);
        dif = fabs(newx - x);
        x = newx;
        count += 1;
    }while(dif > correct && count < MAX_ITER);

    printf("\n\nBY ITERATION METHOD REAL ROOT = %lf",x);
    printf("\n\nNUMBER OF ITERATIONS = %d",count);

    return 0;
}

double phi(double x)
{
    return (log(x) + 7) / 2;
}
```

Using this program solve the following equations:

a. $2x - log_{10}(x) - 7 = 0$

b. $x^2 - log(x) - 10 = 0$

c. $2x - 3sin(x) - 5 = 0$

d. $xlog_{10}(x) - 1.2 = 0$

e. $log(x) = cos(x)$

f. $x - exp(-x) = 0$

(3) Solution of the equation x^2 - log(x) -10 = 0 by Newton Raphson method

The C Program

```
#include
#include
```

```
#define MAX_ITER 100

double f(double INDEPENDENT_VARIABLE);
double df(double INDEPENDENT_VARIABLE);

int main()
{
    int n, count = 0;
    double x, correct;

    printf("FIRST APPROXIMATE ROOT : ");
    scanf("%lf",&x);
    printf("\nCORRECTED TO NUMBER OF DECIMAL PLACES : ");
    scanf("%d",&n);

    correct = pow(10,-(n+1));
    do{
        x = x - f(x)/df(x);
        count += 1;
    }while(fabs(f(x)) > correct && count < MAX_ITER);

    printf("\n\nBY NEWTON-RAPHSON METHOD REAL ROOT = %lf",x);
    printf("\n\nNUMBER OF ITERATION = %d",count);

    return 0;
}

double f(double x)
{
    return x*x - log(x) -10;
}

double df(double x)
{
    double h = 0.0001;
    return (f(x + h) - f(x-h))/(2*h);
}
```

Using this program solve the following equations:

a. $2x - \log_{10}(x) - 7 = 0$
b. $x^2 - \log(x) - 10 = 0$
c. $2x - 3\sin(x) - 5 = 0$
d. $x\log_{10}(x) - 1.2 = 0$
e. $\log(x) = \cos(x)$
f. $x - \exp(-x) = 0$