Numerical Methods HW #06

1. 다음 연립미분방정식을 $0 \le t \le 5$에서 구간간격 h=0.01로 하여 수정 Euler 법으로 풀어라.

$$\frac{dx}{dt} = e^t - 3\frac{dy}{dt} - y \ , \quad x(0) = -1$$

$$\frac{dy}{dt} = y + x \ , \quad y(0) = \frac{5}{4}$$

연립 미분방정식을 풀기 위해 $x = y_1$, $y = y_2$, $\frac{dy}{dt} = y_3$로 놓으면,

$$\frac{dy_1}{dt} = e^t - 3y_3 - y_2$$

$$\frac{dy_2}{dt} = y_3 = y_2 + y_1$$

두 식을 연립하여 y_3를 소거하면

$$\frac{dy_1}{dt} = e^t - 3y_1 - 4y_2$$

$$\frac{dy_2}{dt} = y_2 + y_1$$

이 두 개의 방정식으로 연립 미분방정식을 푼다.

<Code>

```c
#include <stdio.h>
#include <math.h>

int main()
{
	int a=0;
	int b=5;
	int i;
	double h=0.01;
	int n=(b-a)/h;

	double y1[n+1]={0};//x
	double y2[n+1]={0};//y
	y1[0]=-1;y2[0]=(double)5/(double)4;
	//Predictor
	double y1p[n]={0};
	double y2p[n]={0};
	//Corrector
	double y1c[n]={0};
	double y2c[n]={0};
	double f1[n+1]={0};
	double f2[n+1]={0};
	f1[0]=exp(0)-3*y1[0]-4*y2[0];
	f2[0]=y1[0]+y2[0];
	for(i=1;i<=n;i++)
	{
		double t=a+h*i;
		y1p[i-1]=y1[i-1]+h*f1[i-1];
		y2p[i-1]=y2[i-1]+h*f2[i-1];
		f1[i]=exp(t)-3*y1p[i-1]-4*y2p[i-1];
		f2[i]=y1p[i-1]+y2p[i-1];
		y1c[i-1]=y1[i-1]+h/2*(f1[i-1]+f1[i]);
		y2c[i-1]=y2[i-1]+h/2*(f2[i-1]+f2[i]);
```

```
                y1[i]=y1c[i-1];
                y2[i]=y2c[i-1];
        }
        FILE*HW06_1 = fopen("HW06_1.csv","w+");
        fprintf(HW06_1,"t,x,y\n");
        for(i=0;i<=n;i++)
        {
                fprintf(HW06_1,"%lf,%lf,%lf\n",a+h*i,y1[i],y2[i]);
        }
        fclose(HW06_1);
}
```

<Result>

t	y1	y2
0	-1	1.25
0.01	-1.00985	1.252463
0.02	-1.019403	1.254852
0.03	-1.028666	1.25717
0.04	-1.037644	1.259421
0.05	-1.046342	1.261605
... 중간 생략 ...		
4.94	-0.076401	34.985356
4.95	-0.075754	35.336185
4.96	-0.075111	35.690547
4.97	-0.074473	36.048476
4.98	-0.07384	36.410009
4.99	-0.073211	36.775182
5	-0.072587	37.14403

2. 다음 연립방정식을 $0 \le t \le 10$에서 구간간격을 h=0.1로 하여 4계 Runge-Kutta 법으로 풀고, 그 결과를 주어진 엄밀해와 비교하라.

$$\frac{dx}{dt} = 2x - 3y \quad , \quad x(0) = 8$$

$$\frac{dy}{dt} = y - 2x \quad , \quad y(0) = 3$$

엄밀해 : $x(t) = 5e^{-t} + 3e^{4t}$
$y(t) = 5e^{-t} - 2e^{4t}$

연립 미분방정식을 풀기 위해 $x = y_1$, $y = y_2$ 로 놓으면,

$$\frac{dy_1}{dt} = 2y_1 - 3y_2 = f_1$$

$$\frac{dy_2}{dt} = y_2 - 2y_1 = f_2$$

<Code>

```c
#include<stdio.h>
#include<math.h>
double y1e(double t)
{
	return 5*exp(-t)+3*exp(4*t);
}
double y2e(double t)
{
	return 5*exp(-t)-2*exp(4*t);
}
int main()
{
	int a=0;
	int b=10;
	int i;
	double h=0.1;
	int n=(b-a)/h;
	double y1[n+1]={0};
	double y2[n+1]={0};
	y1[0]=8; y2[0]=3;
	double k11,k12,k13,k14,k21,k22,k23,k24;
	for (i=1;i<=n;i++)
	{
		double t=a+i*h;
		k11=2*y1[i-1]-3*y2[i-1];
		k21=y2[i-1]-2*y1[i-1];

		k12=2*(y1[i-1]+h/2*k11)-3*(y2[i-1]+h/2*k21);
		k22=(y2[i-1]+h/2*k21)-2*(y1[i-1]+h/2*k11);

		k13=2*(y1[i-1]+h/2*k12)-3*(y2[i-1]+h/2*k22);
		k23=(y2[i-1]+h/2*k22)-2*(y1[i-1]+h/2*k12);

		k14=2*(y1[i-1]+h*k13)-3*(y2[i-1]+h*k23);
		k24=(y2[i-1]+h*k23)-2*(y1[i-1]+h*k13);

		y1[i]=y1[i-1]+h/6*(k11+2*(k12+k13)+k14);
		y2[i]=y2[i-1]+h/6*(k21+2*(k22+k23)+k24);
	}
	FILE*data=fopen("HW06_2.csv","w+");
	fprintf(data,"t,y1,y1 엄밀해,상대오차1(%%),y2,y2 엄밀해, 상대오차2(%%)\n");
	for(i=0;i<=n;i++)
	{
		double t=a+i*h;
		double e1=fabs((y1[i]-y1e(t))/y1e(t))*100;
		double e2=fabs((y2[i]-y2e(t))/y2e(t))*100;
		fprintf(data,"%lf,%lf,%lf,%lf,%lf,%lf,%lf\n",t,y1[i],y1e(t),e1,y2[i],y2e(t),e2);
	}
	fclose(data);
}
```

<Result>

t	y1	y1 엄밀해	상대오차 1(%)	y2	y2 엄밀해	상대오차 2(%)
0	8	8	0	3	3	0
0.1	8.999387	8.999661	0.003041	1.540721	1.540538	0.011888
0.2	10.76946	10.770277	0.007586	-0.356882	-0.357428	0.152736
0.3	13.662613	13.664442	0.013384	-2.934922	-2.936143	0.041583
0.4	18.207059	18.210698	0.01998	-6.552037	-6.554465	0.037039
0.5	25.193036	25.199822	0.026928	-11.740933	-11.745459	0.038535
0.6	35.801439	35.813587	0.033921	-19.294193	-19.302295	0.041971
... 중간 생략 ...						
9.4	6.36933E+16	6.4061E+16	0.574051	-4.24622E+16	-4.27073E+16	0.574051
9.5	9.50134E+16	9.55678E+16	0.58014	-6.33422E+16	-6.37119E+16	0.58014
9.6	1.41735E+17	1.4257E+17	0.586229	-9.44897E+16	-9.50469E+16	0.586229
9.7	2.1143E+17	2.1269E+17	0.592317	-1.40953E+17	-1.41793E+17	0.592317
9.8	3.15398E+17	3.17296E+17	0.598405	-2.10265E+17	-2.11531E+17	0.598405
9.9	4.70489E+17	4.7335E+17	0.604493	-3.13659E+17	-3.15567E+17	0.604493
10	7.01844E+17	7.06156E+17	0.61058	-4.67896E+17	-4.70771E+17	0.61058

3. 다음 적분형 미분방정식을 0≤t≤10에서 구간간격 h=0.1로 하여 Adams-Moulton with mop-up 법으로 풀고, 그 결과를 엄밀해와 비교하라.

$$y'(t) + 5\int_0^t cos\,2(t-u)y(u)\,du = 10 \quad , \quad y(0) = 2$$

엄밀해 : $y(t) = \frac{1}{27}(24 + 120t + 30\cos 3t + 50\sin 3t)$

<Code>

```
#include<stdio.h>
#include<math.h>
double func(double t)
{
	return (24+120*t+30*cos(3*t)+50*sin(3*t))/27;
}
int main()
{
	int a=0;
	int b=10;
	int i;
	double h=0.1;
	int n=(b-a)/h;
	double y1[n+1]={0};
	double y2[n+1]={0};
	y1[0]=2;
	y2[0]=0;
	double t[n+1]={0};
	t[0]=a;
	double f1[n+1]={0};
	double f2[n+1]={0};
	f1[0]=10-5*y2[0];
	f2[0]=y1[0]-2;
```

```
	double k11,k12,k13,k14,k21,k22,k23,k24;
	for(i=0;i<=n;i++)
	{
		t[i]=a+i*h;
	}
	//Runge-Kutta
	for (i=1;i<=3;i++)
	{
		k11=-5*y2[i-1]+10;
		k21=y1[i-1]-2*cos(2*t[i-1]);

		k12=-5*(y2[i-1]+h/2*k21)+10;
		k22=(y1[i-1]+h/2*k11)-2*cos(2*(t[i-1]+h/2));

		k13=-5*(y2[i-1]+h/2*k22)+10;
		k23=(y1[i-1]+h/2*k12)-2*cos(2*(t[i-1]+h/2));

		k14=-5*(y2[i-1]+h*k23)+10;
		k24=(y1[i-1]+h*k13)-2*cos(2*(t[i-1]+h));

		y1[i]=y1[i-1]+h/6*(k11+2*(k12+k13)+k14);
		y2[i]=y2[i-1]+h/6*(k21+2*(k22+k23)+k24);

		f1[i]=-5*y2[i]+10;
		f2[i]=y1[i]-2*cos(2*t[i]);
	}
	//Adams-Moulton
	double yp1[n+1]={0};
	double yp2[n+1]={0};//Predictor
	double yc1[n+1]={0};
	double yc2[n+1]={0};//Corrector
	double fp1[n+1]={0};
	double fp2[n+1]={0};
	for(i=4;i<=n;i++)
	{
		yp1[i]=y1[i-1]+h/24*(55*f1[i-1]-59*f1[i-2]+37*f1[i-3]-9*f1[i-4]);
		yp2[i]=y2[i-1]+h/24*(55*f2[i-1]-59*f2[i-2]+37*f2[i-3]-9*f2[i-4]);

		fp1[i]=-5*yp2[i]+10;
		fp2[i]=yp1[i]-2*cos(2*t[i]);

		yc1[i]=y1[i-1]+h/24*(9*fp1[i]+19*f1[i-1]-5*f1[i-2]+f1[i-3]);
		yc2[i]=y2[i-1]+h/24*(9*fp2[i]+19*f2[i-1]-5*f2[i-2]+f2[i-3]);

		f1[i]=-5*yc2[i]+10;
		f2[i]=yc1[i]-2*cos(2*t[i]);

		y1[i]=yc1[i];
		y2[i]=yc2[i];
	}
	FILE* data=fopen("HW06_3.csv","w+");
	fprintf(data,"t,엄밀해, 수치해, 상대오차(%%)\n");
	for(i=0;i<=3;i++)
	{
		fprintf(data,"%lf,%lf,%lf,%lf\n",t[i],func(t[i]),y1[i],fabs((y1[i]-
func(t[i]))/func(t[i]))*100);
	}
	for(i=4;i<=n;i++)
	{
		fprintf(data,"%lf,%lf,%lf,%lf\n",t[i],func(t[i]),yc1[i],fabs((yc1[i]-
func(t[i]))/func(t[i]))*100);
	}
	fclose(data);
}
```

<Result>

t	엄밀해	수치해	상대오차(%)
0	2	2	0
0.1	2.942078	2.9915	1.679839
0.2	3.740452	3.93132	5.10281
0.3	4.363505	4.766809	9.242652
0.4	4.795285	5.446957	13.589866
0.5	5.036921	5.925079	17.632948
... 중간 생략 ...			
9.5	41.61381	18.808257	54.802848
9.6	41.665216	18.569887	55.430719
9.7	41.885481	17.500045	58.219306
9.8	42.29463	15.643348	63.013393
9.9	42.895815	13.080229	69.506982
10	43.675036	9.9234	77.27901

4. 다음 주어진 Stiff 미분방정식을

$$y' = f(t,y) = -500y + 2000 - 1000e^{-2t}, \quad y(0) = 0$$

Euler 양함수법과 Euler 음함수법으로 0≤t≤1 사이에서 풀어 엄밀해와 그 결과를 비교하라. 여기서 구간 h=0.01, 0.0001로 택하라. 단, 이 식의 엄밀해는

$$y(t) = 4 - \frac{1000}{498}e^{-2t} - \frac{992}{498}e^{-500t}$$

이다.

(a) Euler 양함수법

<Code>

```c
#include <stdio.h>
#include <math.h>
double func(double t)
{
	return 4-1000/498*exp(-2*t)-992/498*exp(-500*t);
}
int main()
{
	int i;
	int a=0;
	int b=1;
	double h=0.01; //or h=0.0001
	int n=(b-a)/h;
	double y[n+1]={0};
	y[0]=0;
	double f[n]={0};
	for(i=1;i<=n;i++)
	{
		double t=a+i*h;
		f[i-1]=-500*y[i-1]+2000-1000*exp(-2*(a+(i-1)*h));
		y[i]=y[i-1]+h*f[i-1];
	}

	FILE* data=fopen("HW06_4_1_b.csv","w+");
	fprintf(data,"t,수치해,엄밀해,상대오차(%%)\n");
	for(i=0;i<=n;i++)
	{
		double t=a+i*h;
		double error=fabs((func(t)-y[i])/func(t));
		fprintf(data,"%lf,%lf,%lf,%lf\n",t,y[i],func(t),error*100);
	}
	fclose(data);
}
```

<Result>

t	수치해	엄밀해	상대오차(%)	t	수치해	엄밀해	상대오차(%)
0	0	1	100	0	0	1	100
0.01	10	2.032865	391.916652	0.0001	0.1	1.049171	90.468661
0.02	-29.801987	2.078376	1533.907566	0.0002	0.19502	1.095962	82.205594
0.03	129.600053	2.116471	6023.40426	0.0003	0.285309	1.140492	74.983685
0.04	-507.817855	2.153767	23678.12073	0.0004	0.371104	1.182869	68.626818
0.05	2042.040259	2.190325	93130.00494	0.0005	0.452628	1.223198	62.996323
... 중간 생략 ...				... 중간 생략 ...			
0.95	3.12607E+57	3.700863	8.44687E+58	0.9995	3.727971	3.729059	0.029176
0.96	-1.25043E+58	3.706786	3.37335E+59	0.9996	3.728025	3.729113	0.02917
0.97	5.00172E+58	3.712592	1.34723E+60	0.9997	3.728079	3.729167	0.029164
0.98	-2.00069E+59	3.718283	5.38067E+60	0.9998	3.728134	3.729221	0.029158
0.99	8.00274E+59	3.723862	2.14904E+61	0.9999	3.728188	3.729275	0.029151
1	-3.2011E+60	3.729329	8.58357E+61	1	3.728243	3.729329	0.029145

(b) Euler 음함수법

$$f_{n+1} = -500y_{n+1} + 2000 - 1000e^{-2t_{n+1}}$$

$$f_{n+1} = \frac{-500y_n + 2000 - 1000e^{-2t_{n+1}}}{1 + 500h}$$ 를 이용한다.

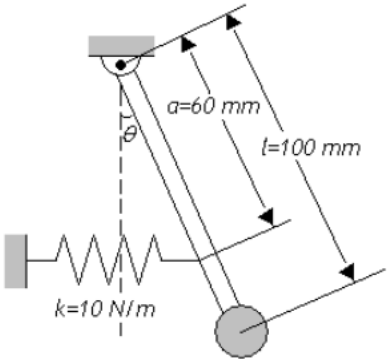
<Code>

```c
#include<stdio.h>
#include<math.h>
double func(double t)
{
	return 4-1000/498*exp(-2*t)-992/498*exp(-500*t);
}
int main()
{
	int i;
	int a=0;
	int b=1;
	double h=0.01; // or h=0.0001
	int n=(b-a)/h;
	double y[n+1]={0};
	y[0]=0;
	double f[n]={0};
	for(i=1;i<=n;i++)
	{
		double t=a+i*h;
		f[i-1]=(-500*y[i-1]+2000-1000*exp(-2*(t+h)))/(1+500*h);
		y[i]=y[i-1]+h*f[i-1];

	}
	FILE* data=fopen("HW06_4_2_b.csv","w+");
	fprintf(data,"t,수치해,엄밀해,상대오차(%%)\n");
	for(i=0;i<=n;i++)
	{
		double t=a+i*h;
		double error=fabs((func(t)-y[i])/func(t));
		fprintf(data,"%lf,%lf,%lf,%lf\n",t,y[i],func(t),error*100);
	}
	fclose(data);
}
```

<Result>

h=0.01				h=0.0001			
t	수치해	엄밀해	상대오차(%)	t	수치해	엄밀해	상대오차(%)
0	0	1	100	0	0	1	100
0.01	1.732018	2.032865	14.79917	0.0001	0.095276	1.049171	90.918904
0.02	2.052395	2.078376	1.250031	0.0002	0.186034	1.095962	83.025472
0.03	2.136872	2.116471	0.963933	0.0003	0.27249	1.140492	76.10768
0.04	2.181416	2.153767	1.28375	0.0004	0.354847	1.182869	70.001107
0.05	2.218702	2.190325	1.295553	0.0005	0.433302	1.223198	64.576281
... 중간 생략 ...				... 중간 생략 ...			
0.95	3.705597	3.700863	0.127912	0.9995	3.728025	3.729059	0.027724
0.96	3.711426	3.706786	0.125179	0.9996	3.728079	3.729113	0.027718
0.97	3.71714	3.712592	0.122508	0.9997	3.728134	3.729167	0.027712
0.98	3.722741	3.718283	0.119899	0.9998	3.728188	3.729221	0.027706
0.99	3.728231	3.723862	0.117348	0.9999	3.728242	3.729275	0.0277
1	3.733613	3.729329	0.114856	1	3.728297	3.729329	0.027694

5. 그림과 같은 진동계의 운동방정식은 다음과 같다.



$$\theta'' + \left(\frac{g}{l} + \frac{ka^2}{ml^2}\right)\theta = 0$$

$$\theta(0) = \frac{\pi}{4}, \qquad \theta'(0) = 0$$

시간 t에 대한 θ의 변위를 구하여라. 단, m = 1kg, g=9.81m/s²이다.

$\frac{g}{l} + \frac{ka^2}{ml^2}$은 상수이므로 $C = \frac{g}{l} + \frac{ka^2}{ml^2} = 38.916$ 이므로,

주어진 운동방정식은 $\theta'' + C\theta = 0$ 으로 나타내어진다.

$\theta = y_1$, $\frac{d\theta}{dt} = y_2$ 로 놓으면 $\frac{dy_1}{dt} = y_2$, $\frac{dy_2}{dt} = -Cy_1$ 로 주어진 방정식을 나타낼 수 있다.

Runge-Kutta 4계 방법으로 $0 \le t \le 10$에서의 해를 h=0.1로 구한다.

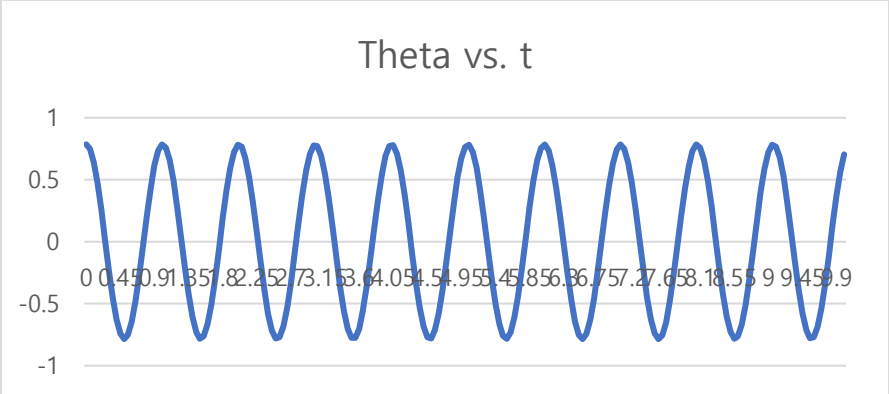
<Code>

```
#include <stdio.h>
#include <math.h>
int main()
{
	int i;
	int a=0;
	int b=10;
	double h=0.05;
	int n=(b-a)/h;
	double y1[n+1];
	double y2[n+1];
	double c=38.916;
```

```
        y1[0]=M_PI/4;
        y2[0]=0;
        double t[n+1]={0};
        t[0]=a;
        double k11,k21,k31,k41,k12,k22,k32,k42;
        for(i=1;i<=n;i++)
        {
                t[i]=a+i*h;

                k11=y2[i-1];
                k12=-c*y1[i-1];

                k21=y2[i-1]+h/2*k12;
                k22=-c*(y1[i-1]+h/2*k11);

                k31=y2[i-1]+h/2*k22;
                k32=-c*(y1[i-1]+h/2*k21);

                k41=y2[i-1]+h*k32;
                k42=-c*(y1[i-1]+h*k31);

                y1[i]=y1[i-1]+h/6*(k11+2*(k21+k31)+k41);
                y2[i]=y2[i-1]+h/6*(k12+2*(k22+k32)+k42);
                printf("%lf       %lf\n",t[i],y1[i]);
        }
                FILE* data=fopen("HW06_5.csv","w+");
        fprintf(data,"t,theta\n");
        for(i=0;i<=n;i++)
        {
                fprintf(data,"%lf,%lf\n",t[i],y1[i]);
        }
        fclose(data);
}
```

<Result>

t	theta
0	0.785398
0.05	0.747502
0.1	0.637481
0.15	0.465952
0.2	0.249466
0.25	0.008912
0.3	-0.232499
0.35	-0.451473
0.4	-0.626883
0.45	-0.741803
0.5	-0.785146
0.55	-0.752731
0.6	-0.647687
0.65	-0.480149
0.7	-0.266285
0.75	-0.02673



*진동운동에 대한 해를 graphical 하게 구할 수 있다.