

HW #3
February 13, 2006
1)

a)

```
#include <stdio.h>

int add(int n);
int main()
{
    int n;
    printf("Enter n = ");
    scanf("%d", &n);
    printf("%d factorial =%d \n", n, add(n));
    return 0;
}

int add(int n)
{
    if (n==1) return 1;
    return n*add(n-1);
}
```

```
[rtt5188@omega rtt5188]$ a.out
Enter n = 21
21 factorial =-1195114496
```

b)

```
#include <stdio.h>

int main()
{
    int i;
    int n;
    int sum;
    sum=1;
    printf("Enter n = ");
    scanf("%d", &n);
    for(i=1;i<=n;i++)
    {
        sum=i*sum;
    }
    printf("%d factorial = %d \n",n,sum);
    return 0;
}
```

```
[rtt5188@omega rtt5188]$ a.out
Enter n = 11
11 factorial = 39916800
```

2)

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

hw3

```
int main()
{
int i;
int n,b;
float sum=1.0, master_sum=1.0;
printf("Enter n = \n");
scanf("%d", &n);
for(i=1;i<=n;i++)
{
    for(b=1;b<=i;b++)
    {
        sum=b*sum;
    }
    master_sum=master_sum+(1.0/sum);
    sum=1.0;
}
printf("The sumation 1/%d! = %f \n",n,master_sum);
return 0;
}
```

```
[rtt5188@omega rtt5188]$ a.out
Enter n =
11
The sumation 1/11! = 2.718282
[rtt5188@omega rtt5188]$ a.out
Enter n =
35
The sumation 1/35! = 2.718282
```

3)

```
#include <stdio.h>

int mysequence(int n)
{
if (n==0) return 2;
if (n==1) return 3;
return 2*mysequence(n-1)-3*mysequence(n-2);
}

int main()
{
int n;
printf("Enter n = ");
scanf("%d", &n);
n=n+2;
printf("%d\n",mysequence(n));
return 0;
}
```

```
[rtt5188@omega rtt5188]$ a.out
Enter n = 20
-95436
```

4)

```
#include <stdio.h>
int main()
```

hw3

```

{
int i;

float a[100]={6.2809, 2.68913, 8.54087, 0.291223, 6.36043, 1.09746,
9.40927, 8.84139, 1.15765, 5.30968, 8.78946, 5.55576, 2.49733, 9.11668, 4.71846$
0.0468933, 4.59108, 7.3383, 3.7373, 4.4102, 3.71414, 6.66195, 4.6776,
8.1293, 1.02501, 8.12109, 4.38637, 1.76887, 9.92756, 8.71182, 5.54498, 0.611219$
4.61788, 9.92236, 9.98922, 8.11389, 5.5012, 5.2039, 6.07206, 8.06699,
0.910121, 7.8656, 2.33477, 3.65679, 7.19598, 1.20364, 7.65717, 5.52749,
6.17097, 3.08256, 3.2708, 3.75863, 6.24341, 4.37074, 7.72581, 3.14741,
1.62553, 4.44837, 7.73659, 5.03352, 6.12433, 9.24447, 1.66453, 6.96653,
5.21421, 1.37887, 9.32976, 3.30974, 8.01823, 0.175227, 1.67259, 7.78224,
1.84726, 7.09267, 8.40179, 4.02362, 5.60385, 2.72194, 0.675983, 0.876211,
3.97832, 8.27356, 2.93939, 5.84269, 7.85399, 9.02909, 1.27486, 8.87616,
2.63978, 7.65022, 1.9451, 5.56643, 4.62156, 7.475, 0.272512, 7.78419,
2.7743, 0.382325, 1.87072};
float sum=0.0;

for (i=0;i<100;i++)
    sum = sum +a[i];

printf("The average of the 100 numbers is = %f\n", sum/100);
return 0;
}

```

```

[rtt5188@omega rtt5188]$ a.out
The average of the 100 numbers is = 5.016042

```

```

5)
a)
-19, 1
-27, 17
30, 11

b)
4, -8,-4,12
-2, 4,-2,-6
1, -2,-1, 3
6,-12,-6,18

```

```

6)

#include <stdio.h>
int main()

{

float a[20][5]={{1.5615, 3.72505, 1.17518, -1.58305, -1.28521}, {-1.82493,
1.05322, 2.50276,
1.31591, 2.4593}, {1.26372, 3.43518, -0.276327,
0.167821, -0.641508}, {2.30645, 1.12868, -1.64937, -0.901803,
3.1905}, {3.01095,
1.53186, -0.229594, -3.64928, -2.55055}, {1.80681, -3.05441, 1.93377,
2.73467, -0.368258}, {-0.107631, 3.43101, -2.58125, 1.17244,
2.62865}, {3.99583, 1.69508, -2.99538, -0.729839, -2.31062}, {-3.4336,
2.65399, -3.82804, -1.50112, -2.44455}, {-2.87788,
2.401557, -1.85183, -3.89401, -0.684686}, {-0.544033, 0.214401, -2.62867,
3.68357, 3.5636}, {0.78339, 3.95257, -1.48887, -3.06505,
0.787563}, {-1.74251, -2.49349, 1.66479, -0.901815, -2.30891}, {-1.14748,
1.49282, -3.4007, -3.86435, -2.26961}, {-2.90873, 2.45113, -3.97034,
-2.41508, 1.6353}, {-1.76327, 2.65833, 2.73151, 2.0717, 1.45334}, {2.70575,

```

hw3

```
0.22038, 1.13675, -3.33422, 0.44826}, {-1.28613, 3.47197,  
1.5676, -1.24283, 3.86135}, {-2.02085, 8.968294, -1.37848,  
2.13096, -13.11212}, {2.51716, -1.40814, 3.71588, -0.747417, 0.280426}},
```

```
b[5][10]={{-0.066466, -3.01563, 1.18088, 2.82708, 1.22778, 0.763991, -3.95587,  
2.1613, -3.22048, -1.94988}, {3.42784, -3.40629, 2.02236, -1.81124,  
2.59301, -0.374589, -0.599161, 0.0578041, 1.70513, 1.10825}, {-13.19102,  
0.341925, -1.54745, -3.17217, 0.875443, -0.642447, 1.27167, -1.99926,  
3.64766, 2.59356}, {1.22754, -0.160558, 2.86814, 0.543444,  
0.655377, -0.754264, -3.15421, -1.64532, 2.06236, 3.62032}, {1.44495,  
2.29688, -3.64277, -1.48793, 0.63597, 2.04505, 1.90468, -2.31575,  
3.76053, 2.5974}},
```

```
c[20][10];
```

```
int i,j,k;
```

```
/* initialization of c*/
```

```
for (i=0;i<20;i++) for(j=0;j<10;j++) c[i][j]=0;
```

```
for (i=0;i<20;i++)  
  for (j=0;j<10;j++)  
    for (k=0;k<5;k++)  
      c[i][j]=c[i][j]+a[i][k]*b[k][j];
```

```
for (i=0; i<20; i++)  
{  
  for (j=0; j<10 ; j++)  
    printf("%10.3f", c[i][j]);  
  printf("\n");  
}
```

```
return 0;  
}
```

```
[rtt5188@omega rtt5188]$ a.out  
-6.637 -19.693 7.700 -5.008 10.750 -2.392 -4.369 6.822  
-2.488 -4.938  
-24.113 8.209 -9.082 -17.950 5.108 0.640 10.304 -16.747  
28.764 22.368  
14.615 -17.107 11.685 -0.727 9.919 -1.582 -9.160 4.692  
-1.287 -0.432  
28.976 -3.891 -6.650 4.471 5.753 9.604 -2.976 2.443  
-1.382 -2.502  
-0.086 -19.649 5.833 8.278 3.454 -0.589 -6.468 18.966  
-25.040 -24.605  
-33.274 4.332 2.149 6.540 -2.451 -1.534 -12.185 -3.784  
0.282 7.051  
51.055 -6.396 4.593 -1.605 8.945 4.782 -3.604 -2.890  
9.085 8.390  
40.822 -24.038 19.106 20.770 4.731 0.167 -22.731 21.274  
-31.099 -22.325  
54.446 -5.369 11.836 0.451 -3.224 -5.025 7.203 8.516  
-10.669 -12.076  
27.082 -1.082 -4.350 -7.709 -1.915 -0.372 18.569 5.614  
-3.997 -12.406  
45.117 7.605 1.443 3.112 2.267 5.702 -6.150 -10.221  
13.527 17.073  
30.512 -14.034 -0.437 -3.059 8.400 3.997 3.807 8.117  
-4.574 -10.059  
-34.835 9.159 -3.852 -2.745 -9.207 -5.508 8.951 -0.408
```

		hw3						
-3.110	-4.310							
42.029	-7.380	4.111	6.117	-4.491	-0.978	7.186	16.019	
-22.668	-24.813							
60.367	3.209	-5.218	-3.814	-1.234	4.576	15.721	1.979	
0.233	-6.405							
-22.159	0.202	-0.285	-19.501	9.401	-2.688	5.090	-15.892	
29.913	24.744							
-17.864	-6.957	-9.314	1.165	2.989	4.686	1.981	8.036	
-9.382	-12.990							
-4.638	1.657	-14.554	-21.318	10.437	5.544	16.276	-12.610	
27.738	15.951							
32.729	-55.385	71.760	3.084	12.625	-32.440	-30.828	25.765	
-28.142	-16.038							
-54.523	-0.760	-8.791	-2.944	2.381	1.201	-1.497	-1.490	
2.560	1.191							