

Practice 05. Functions

Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National University

<http://bi.snu.ac.kr>

recursive functions

```
// 0 + 1 + 2 + ... + n
int sum(int n)
{
    if (n <= 0)
        return 0;
    return n + sum(n - 1)
}
```

recursive functions

```
// n-th value of the Fibonacci sequence (0, 1, 1, 2, 3, 5, 8, 13, ...)
int fibo(int n)
{
    if (n <= 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return fibo(n - 1) + fibo(n - 2);
}
```

recursive functions

```
// a^b, assume b >= 0
int expo(int a, int b)
{
    if (b == 0)
        return 1;
    else if (b == 1)
        return a;
    else if (b == 2)
        return a*a;
    else if (b % 2 == 0) // b is even
        return expo( expo(a, b/2), 2);
    else // b is odd
        return a * expo( expo(a, (b-1)/2), 2);
}
```

Practice Submission

- **Submit the practice problems if they are not checked in the class time.**
- Submit the solution code of **practice problem 01, 02** by email.
- hnikwak@bi.snu.ac.kr
- Mail title: **prg_[student number]_practice05**
 - prg_2014-12345_practice05
- Submit two source files named **p01.c, p02.c**, for each problem.
- Due to : **4/8(Wed) 23:59 pm**

Assignment Submission

- Create a directory named **assignment** in you home directory.
- Create a directory named **05** in you **assignment** directory.
- Put your C files named **p[# of problem].c** for each problem.
 - p01.c
 - p02.c
 - ...
- Due to : **4/8(Wed) 23:59 pm**

practice 01 – inverse matrix

- Complete the following code.

```
int main(void)
{
    int n; // the number of matrix
    int i;
    scanf("%d", &n);

    for (i = 0; i < n; i++)
    {
        double a11, a12, a21, a22; // elements of a 2*2 matrix
        scanf("%lf%lf%lf%lf", &a11, &a12, &a21, &a22);
    }
}
```

```
if (no_inverse(a11, a12, a21, a22))
    printf("invalid matrix\n");
else
{
    // elements of the inverse matrix
    double b11, b12, b21, b22;
    b11 = inv_11(a11, a12, a21, a22);
    b12 = inv_12(a11, a12, a21, a22);
    b21 = inv_21(a11, a12, a21, a22);
    b22 = inv_22(a11, a12, a21, a22);
    printf("%f %f %f %f\n", b11, b12, b21, b22);
}
}
return 0;
}
```


[Input]

5

1 2 1 2

1 2 3 4

5.4 2.2 6.4 1.2

1 1 0 1

10.52 7.12 29.5 -52.4

[Output]

invalid matrix

-2.000000 1.000000 1.500000 -0.500000

-0.157895 0.289474 0.842105 -0.710526

1.000000 -1.000000 -0.000000 1.000000

0.068831 0.009353 0.038750 -0.013819

practice 02 – recursive function

- The input contains two integers n, k ($0 < n < 10, 0 \leq k \leq n$)

- Output $\binom{n}{k}$.

- Use the recursive function. Note that

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \text{ and } \binom{n}{n} = \binom{n}{0} = 1$$

[Input]

7 3

[Output]

35

assignment 01 – prime numbers

- Complete the following code.

```
int main(void)
{
    int p = 2;
    // print all prime numbers less than 100
    for(; p < 100; p = next_prime(p))
        printf("%d\n", p);
    return 0;
}
```

assignment 02 – recurrence relation

- $\begin{cases} a_n = a_{n-1} - b_{n-1} \\ b_n = 2a_{n-1} + b_{n-1} \end{cases} \quad (a_1 = 1, b_1 = 1)$
- Output the a_n for a given integer n in the input. ($1 \leq n \leq 10$)

[Input]

4

[Output]

-6