

課題6-1

a

base case: 再帰関数において「再帰呼び出しせずに停止する場合」

b

1

main()

- main
- 仮引数の個数: 0
- 仮引数の型: 該当なし
- 戻り値の型: void

sphereSurface()

- 関数名: sphereSurface
- 仮引数の個数: 1
- 仮引数の型: double
- 戻り値の型: double

square()

- square
- 仮引数の個数: 1
- 仮引数の型: double
- 戻り値の型: double

2

```
3 回.  
main: 引数なし  
sphereSurface: 2  
square: 2
```

課題6-2

実行開始時:
(なし)

「int main(void)」の実行後:
(なし)

(なし)

「printf("%d\n", fact(3));」の実行後:
ああああああああ!ああああああああ!ああああ!ああああああ
(なし)

(なし)

「int fact(int n)」の実行後:

(なし)

(なし)

(なし)

「if (n == 0)」の実行後:

(なし)

(なし)

n : 3

「int fact(int n)」の実行後:

(なし)

(なし)

n : 3

(なし)

「if (n == 0)」の実行後:

(なし)

(なし)

n : 3

n : 2

「int fact(int n)」の実行後:

(なし)

(なし)

n : 3

n : 2

(なし)

「if (n == 0)」の実行後:

(なし)

(なし)

n : 3

n : 2

n : 1

「int fact(int n)」の実行後:

(なし)

(なし)

n : 3

n : 2

n : 1

(なし)

```
「if (n == 0)」の実行後：
(なし)
-----
(なし)
-----
n : 3
-----
n : 2
-----
n : 1
-----
n : 0

「n * fact(n - 1)」の実行後：
(なし)
-----
(なし)
-----
n : 3
-----
n : 2
-----
n : 1

「n * fact(n - 1)」の実行後：
(なし)
-----
(なし)
-----
n : 3
-----
n : 2

「n * fact(n - 1)」の実行後：
(なし)
-----
(なし)
-----
n : 3

「n * fact(n - 1)」の実行後：
(なし)
-----
(なし)

「return 0」の実行後：
(なし)
```

課題6-3

a

関数 f は、引数として受け取った非負整数 n それ自身を返す。

b

関数 f は、引数として受け取った非負整数 n が偶数ならば 0, 奇数ならば 1 を返す。

課題6-4

fib.c のフルパス

```
/home/username/assignments/ProIAJ/06/fib.c
```

fib.c のコード

```
#include <stdio.h>

int memo[1000];

int fib(int n) {
    if (memo[n] != 0) return memo[n];
    if (n == 0) return 0;
    if (n == 1) return 1;
    memo[n] = fib(n - 1) + fib(n - 2);
    return memo[n];
}

int main() {
    printf("fib(3): %d, fib(30): %d, fib(6250): %d\n", fib(3), fib(30), fib(599));
    return 0;
}
```

実行結果

コンパイル

```
username@unix01:~/assignments/ProIAJ/06% cc -o fib fib.c
```

入出力

```
username@unix01:~/assignments/ProIAJ/06$ ./fib
fib(3): 2, fib(30): 832040, fib(6250): 1701251697
username@unix01:~/assignments/ProIAJ/06$
```

課題6-5

binary.c のフルパス

```
/home/username/assignments/ProIAJ/06/binary.c
```

binary.c のコード

```
#include <stdio.h>

int print_binary(int n) {
    if (n == 0) return 0;
    print_binary(n/2);
    printf("%d", n % 2);
    return 0;
}

int main() {
```

```
    print_binary(2000);  
    return 0;  
}
```

実行結果

コンパイル

```
username@unix01:~/assignments/ProIAJ/06% cc -o binary binary.c
```

入出力

```
username@unix01:~/assignments/ProIAJ/06$ ./binary  
11111010000  
username@unix01:~/assignments/ProIAJ/06$
```

課題6-6

ack.c のフルパス

```
/home/username/assignments/ProIAJ/06/ack.c
```

ack.c のコード

```
#include <stdio.h>  
  
long long int memo[1000][1000];  
  
long long int ack(int m, int n) {  
    printf("ack(%d, %d)\n", m, n);  
    if (memo[m][n] != 0) return memo[m][n];  
    if (m == 0) {  
        memo[m][n] = n + 1;  
    } else if (n == 0) {  
        memo[m][n] = ack(m - 1, 1);  
    } else {  
        memo[m][n] = ack(m - 1, ack(m, n - 1));  
    }  
    return memo[m][n];  
}  
  
int main() {  
    int m, n;  
    printf("ack(3, 3): %lld\n", ack(m, n));  
    printf("ack(3, 4): %lld\n", ack(m, n));  
    return 0;  
}
```

実行結果

コンパイル

```
username@unix01:~/assignments/ProIAJ/06% cc -o ack ack.c
```

| 入出力

```
username@unix01:~/assignments/ProIAJ/06$ ./ack
ack(3, 3): 61
ack(3, 4): 125
username@unix01:~/assignments/ProIAJ/06$
```