

プログラミング演習I 課題ドキュメント

b162392

構築: Doxygen 1.8.6

2016 年 11 月 17 日 (木) 14 時 41 分 07 秒

Contents

1	ファイル詳解	1
1.1	functions.c ファイル	1
1.1.1	関数詳解	1
1.2	functions.c	2
1.3	main.c ファイル	3
1.3.1	詳解	3
1.3.2	関数詳解	3
1.4	main.c	5

1 ファイル詳解

1.1 functions.c ファイル

関数

- int `Fibonacci` (int *n*)
n についてのフィボナッチ数を計算する
- int `factorial` (int *n*)
n の階乗を計算する

1.1.1 関数詳解

1.1.1.1 int `factorial` (int *n*)

n の階乗を計算する

引数

<i>n</i>	正の整数
----------	------

戻り値

n の階乗

日付

2016/11/10

著者

佐伯雄飛, B162392

`functions.c` の 行目に定義があります。

```
00036         {
00037     int m;
00038     int f3 = 1;
00039
00040     for (m = 1; m <= n; ++m) {
00041         f3 = f3 * m;
00042     }
00043
00044     return f3;
00045 }
```

1.1.1.2 int Fibonacci (int n)

n についてのフィボナッチ数を計算する

フィボナッチ数 F_n は漸化式 $F_n = F_{n-1} + F_{n-2}$ で定義される . ただし $F_1 = F_2 = 1$.

引数

n	正の整数
-----	------

戻り値

フィボナッチ数 (n)

日付

2016/11/10

著者

佐伯雄飛 , B162392

[functions.c](#) の 行目に定義があります。

```
00014         {
00015     int i;
00016     int f0 = 0;
00017     int f1 = 1;
00018     int f2 = 1;
00019
00020     for (i = 1; i < n; i++) {
00021         f2 = f1 + f0;
00022         f0 = f1;
00023         f1 = f2;
00024     }
00025
00026     return f2;
00027 }
```

1.2 functions.c

```
00001
00002 /** @fn int Fibonacci(int n)
00003  * @brief n についてのフィボナッチ数を計算する
00004  *
00005  * フィボナッチ数 \f$ F_n \f$は漸化式
00006  * \f$ F_n = F_{n-1} + F_{n-2} \f$で定義される .
00007  * ただし \f$ F_1 = F_2 = 1 \f$ .
00008  *
00009  * @param n 正の整数
00010  * @return フィボナッチ数 (n)
00011  * @date 2016/11/10
00012  * @author 佐伯雄飛 , B162392
00013  */
00014 int Fibonacci(int n) {
00015     int i;
00016     int f0 = 0;
00017     int f1 = 1;
00018     int f2 = 1;
00019
00020     for (i = 1; i < n; i++) {
00021         f2 = f1 + f0;
00022         f0 = f1;
00023         f1 = f2;
00024     }
00025
00026     return f2;
00027 }
00028
00029 /** @fn int factorial(int n)
00030  * @brief n の階乗を計算する
00031  * @param n 正の整数
00032  * @return n の階乗
```

```
00033 * @date 2016/11/10
00034 * @author 佐伯雄飛, B162392
00035 */
00036 int factorial(int n) {
00037     int m;
00038     int f3 = 1;
00039
00040     for (m = 1; m <= n; ++m) {
00041         f3 = f3 * m;
00042     }
00043
00044     return f3;
00045 }
```

1.3 main.c ファイル

再帰関数を計算する

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

関数

- int [Fibonacci](#) (int n)
n についてのフィボナッチ数を計算する
- int [factorial](#) (int a)
n の階乗を計算する
- int [main](#) (void)
再帰関数を計算する

1.3.1 詳解

再帰関数を計算する

日付

2016/11/10

著者

佐伯雄飛
B162392

[main.c](#) に定義があります。

1.3.2 関数詳解

1.3.2.1 int factorial (int n)

n の階乗を計算する

引数

n	正の整数
---	------

戻り値

n の階乗

日付

2016/11/10

著者

佐伯雄飛, B162392

[functions.c](#) の 行目に定義があります。

```
00036             {
00037     int m;
00038     int f3 = 1;
00039
00040     for (m = 1; m <= n; ++m) {
00041         f3 = f3 * m;
00042     }
00043
00044     return f3;
00045 }
```

1.3.2.2 int Fibonacci (int n)

n についてのフィボナッチ数を計算する

フィボナッチ数 F_n は漸化式 $F_n = F_{n-1} + F_{n-2}$ で定義される . ただし $F_1 = F_2 = 1$.

引数

n	正の整数
-----	------

戻り値

フィボナッチ数 (n)

日付

2016/11/10

著者

佐伯雄飛, B162392

[functions.c](#) の 行目に定義があります。

```
00014             {
00015     int i;
00016     int f0 = 0;
00017     int f1 = 1;
00018     int f2 = 1;
00019
00020     for (i = 1; i < n; i++) {
00021         f2 = f1 + f0;
00022         f0 = f1;
00023         f1 = f2;
00024     }
00025
00026     return f2;
00027 }
```

1.3.2.3 int main (void)

再帰関数を計算する

入力 :

- 標準入力には、正の整数が1つ与えられる。

出力：

- 与えられた整数 n のフィボナッチ数と階乗を計算する関数を呼び出し、この順番で標準出力に表示する。
- 数値と数値の間には空白を1つ入れる
- 実数 x が正の整数ではない場合、

```
Error: the number must be positive
```

というエラーを表示し (return 0 で) 終了する。

入力例：

5

出力例：

5 120

入力例：

-2

出力例：

```
Error: the number must be positive
```

日付

2016/11/10

著者

佐伯雄飛, B162392

[main.c](#) の行目に定義があります。

```
00049         {
00050     int n;
00051     scanf("%d", &n);
00052
00053     if (n > 0) {
00054         printf("%d ", Fibonacci(n));
00055         printf("%d\n", factorial(n));
00056     } else if (n <= 0) {
00057         printf("Error: the number must be positive");
00058         printf("\n");
00059     }
00060 }
00061 return 0;
00062 }
```

1.4 main.c

```
00001 /** @file main.c
00002  * @brief 再帰関数を計算する
00003  * @date 2016/11/10
00004  * @author 佐伯雄飛
00005  * @author B162392
00006 */
00007
00008 #include <stdio.h>
```

```
00009
00010 int Fibonacci(int n);
00011 int factorial(int a);
00012
00013 /** @fn int main(void)
00014  * @brief 再帰関数を計算する
00015  *
00016  * 入力:
00017  * - 標準入力には、正の整数が 1 つ与えられる。
00018  *
00019  * 出力:
00020  * - 与えられた整数 n のフィボナッチ数と階乗を計算する
00021  *   関数を呼び出し、
00022  *   この順番で標準出力に表示する。
00023  * - 数値と数値の間には空白を 1 つ入れる
00024  * - 実数 x が正の整数ではない場合、
00025  * \verbatim
00026 Error: the number must be positive
00027 \endverbatim
00028  *   というエラーを表示し (return 0 で) 終了する。
00029  *
00030  * 入力例:
00031  * \verbatim
00032 5
00033 \endverbatim
00034  * 出力例:
00035  * \verbatim
00036 5 120
00037 \endverbatim
00038  * 入力例:
00039  * \verbatim
00040 -2
00041 \endverbatim
00042  * 出力例:
00043  * \verbatim
00044 Error: the number must be positive
00045 \endverbatim
00046  * @date 2016/11/10
00047  * @author 佐伯雄飛, B162392
00048  */
00049 int main(void) {
00050     int n;
00051     scanf("%d", &n);
00052
00053     if (n > 0) {
00054         printf("%d ", Fibonacci(n));
00055         printf("%d\n", factorial(n));
00056     } else if (n <= 0) {
00057         printf("Error: the number must be positive");
00058         printf("\n");
00059     }
00060 }
00061 return 0;
00062 }
```