

Training 6

関数

株式会社イーシーエス 出版事業推進委員会

Lesson1 関数の仕組み



Point◆◇関数の仕組みを理解しよう!!

プログラムを開発するときは、機能ごとに小さな単位に分けて開発するのが一般的です。小さな単位に分けるには「関数」を使用します。関数の作成方法や使用方法を練習しましょう。

【問題1】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

変数 a - 変数 b の値を計算して表示する関数 func を使用する。

```
#include <stdio.h>
void func(int, int);

void main(void)
{
    int a = 50;
    int b = 10;

    □① /* 関数 func の呼び出し */
}

void □②
{
    int total;

    total = a - b;
    printf("a-b= %d\n", total);
}
```

【問題2】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

二乗を計算する関数 func を使用してその結果を表示する。

```
#include <stdio.h>
int func(int);

void main(void)
{
    int sq_val;
    int a = 5;

    □① /* 関数 func の呼び出し */

    printf("a*a= %d\n", sq_val);
}

□②
{
    □③ a * a;
}
```

【問題3】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

100 ÷ [整数入力値] を計算して表示する。

※ただし、入力値が0の場合は計算・表示をしないものとする。

```
#include <stdio.h>
void func(int);

void main(void)
{
    int a;

    scanf("%d", &a);

    □①
}

□② func(int x)
{
    int y;

    if( x == 0 )
    {
        □③;
    }
    y = 100 / x;
    printf("100÷%dは%dです。¥n", x, y);
}
```

【問題4】 次のプログラムの問題点を挙げなさい。

```
#include <stdio.h>
double x = 0.0;

void main(void)
{
    int i;

    for( i = 0; i < 5; i++ )
    {
        sum(0.5);
    }

    printf("sum : %lf¥n", x);
}

void sum(double a)
{
    x += a;
}
```

【問題5】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

入力された整数値の絶対値を求める。

```
#include <stdio.h>
#include <stdlib.h>
□①;

void main(void)
{
    int x;
    int y;

    scanf("%d", &x);
    y = □②;
    printf("%dの絶対値は%dです。¥n", x, y);
}

int abs_check(□③)
{
    a = abs(a); /* 変数 a を絶対値化 */
    return a;
}
```

【問題6】 次のプログラムを実行した場合に表示される実行結果を答えなさい。

main.c

```
#include <stdio.h>
int func1(int a, int b);
int func2(int a, int b);

void main(void)
{
    int x = 1;
    int y = 2;

    x = func1(x, y);
    y = func2(x, y);

    printf("x=%d, y=%d¥n", x, y);
}

int func1(int a, int b)
{
    int y;
    y = a + b;
    return y;
}
```

sub.c

```
int func3(int a);

int func2(int a, int b)
{
    int n;
    a = a + b;
    n = func3(a-1);
    return n;
}

int func3(int a)
{
    return a * 10;
}
```

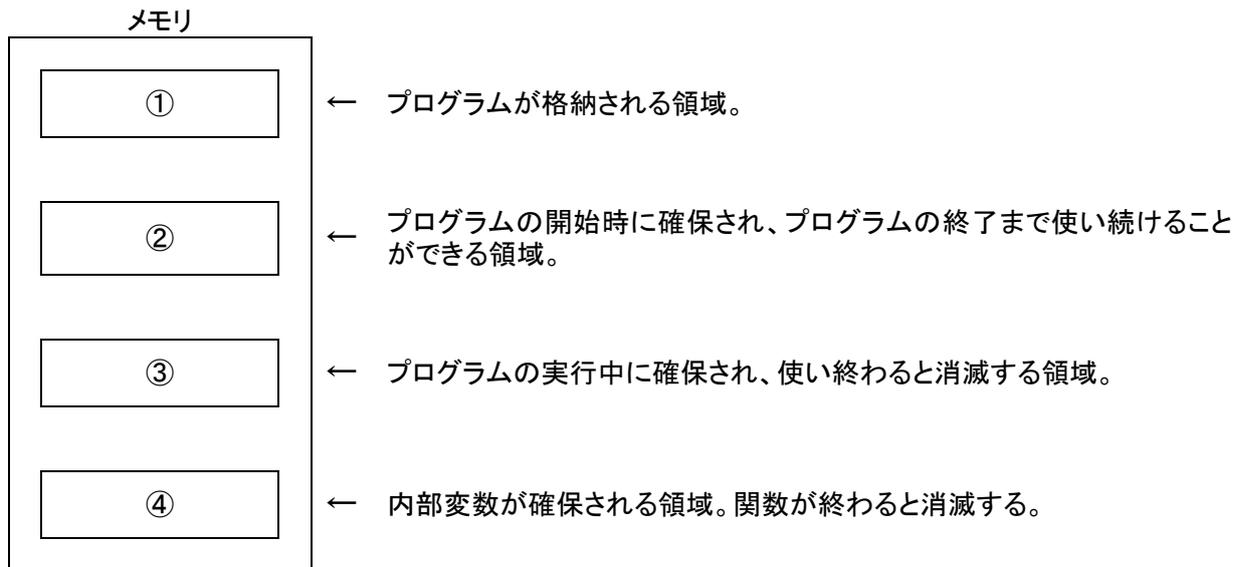
Lesson2 メモリ



Point ◆◇メモリを理解しよう!!

コンピュータのメモリは大きく分けると4種類の領域があり、用途により使い分けられています。

【問題1】 次の①～④に当てはまるメモリの領域を解答群から選びなさい。



解答群

ア. プログラム領域

イ. 動的記憶領域(ヒープ領域)

ウ. 静的記憶領域

エ. 動的記憶領域(スタック領域)

【問題2】 次のプログラム(main.c, sub.c)の①～⑦が表す変数またはコードが、どのメモリ領域に当てはまるか解答群から選びなさい。(⑦は、変数nを対象とする。)

main.c

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

```
int a; .....①
```

```
static int d; ....②
```

```
void sub(int n);
```

```
void main(void)
```

```
{
```

```
    static int b; ....③
```

```
    auto int c; ....④
```

```
    a = 5;
```

```
    b = 2;
```

```
    sub( 10 );
```

```
    c = a + b; .....⑤
```

```
    d = a - c;
```

```
}
```

sub.c

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

```
extern int a; .....⑥
```

```
void sub(int n) .....⑦
```

```
{
```

```
    a = a + n;
```

```
}
```

解答群

ア. プログラム領域

イ. 動的記憶領域(ヒープ領域)

ウ. 静的記憶領域

エ. スタック領域

Lesson3 変数のスコープ



Point◆◇変数宣言の種類を把握しよう!!

変数は、記憶クラス指定と宣言位置により、可視範囲(スコープ)や寿命等の性質が決まります。
可視範囲は、変数だけではなく typedef による宣言や関数等にも存在します。
変数の寿命を考慮して変数宣言をしましょう。

【問題1】 次のプログラムを実行した時、①～④の各行終了直後の **y** の値を答えなさい。

```
#include <stdio.h>
int calc(int);

void main(void)
{
    int z;
    int x = 4;
    int y = 2; .....①

    z = calc(x);

    printf("x= %d, y= %d, z= %d\n", x, y, z); .... ②
}

int calc(int x)
{
    int y = 3; .....③

    y *= x; .....④

    return y;
}
```

【問題2】 次のプログラムを実行した時、①～③の各行終了直後の **y** の値を答えなさい。

```
#include <stdio.h>
int calc(int);
int y = 5;

void main(void)
{
    int z;
    int x;

    x = 4; .....①
    y = 2;
    z = calc(x);
    printf("x= %d, y= %d, z= %d\n", x, y, z); ....②

    y = 10;
    y = calc(x);
    printf("x= %d, y= %d, z= %d\n", x, y, z); ....③
}

int calc(int x)
{
    y *= x;

    return y;
}
```

【問題3】 次のプログラムの①～③の変数宣言で宣言された変数の可視範囲を、それぞれ解答群から選びなさい。

mai.n.c

```
01: #include <stdio.h>
02: int addsum(int val);
03: int x = 5; .....①
04:
05: void main(void)
06: {
07:     int i;
08:     int n; .....②
09:
10:     for( i = 0; i < 3; i++ )
11:     {
12:         n = addsum(x);
13:     }
14:
15:     printf("addsum : %d¥n", n);
16: }
```

sub.c

```
01: int addsum(int val);
02:
03: int addsum(int val)
04: {
05:     static int i = 0; .....③
06:     i += val;
07:     return i;
08: }
```

解答群

- ア. mai.n.c と sub.c 全て可視
- イ. mai.n.c は全て可視 sub.c は不可視
- ウ. mai.n.c は不可視 sub.c は全て可視
- エ. mai.n 関数内のみ可視
- オ. addsum 関数内のみ可視

解答

Training6 関数

Lesson1 関数の仕組み

問題 1	①func(a, b); 又は func(b, a); ②func(int a, int b) 又は func(int b, int a)
問題 2	①sq_val = func(a) ②int func(int a) ③return
問題 3	①func(a); ②voi d ③return
問題 4	①関数 sum のプロトタイプ宣言していない
問題 5	①int abs_check(int a); 又は int abs_check(int) ②abs_check(x) ③int a
問題 6	①x=3, y=40

Lesson2 メモリ

問題 1	①ア ②ウ ③イ ④エ
問題 2	①ウ ②ウ ③ウ ④エ ⑤ア ⑥ウ ⑦エ

【解説】問題 2

- ①、②は外部変数のため、静的記憶領域に確保されます。
③は `static`(静的記憶クラス指定子)が設定されているので、静的記憶領域に確保されます。
④は `auto`(スタック記憶クラス指定子)が設定されているので、スタック領域に確保されます。
⑤はプログラムなので、プログラム領域に確保されます。
⑥の `extern`(外部記憶指定子)は、メモリの確保はされません。この指定子をつけた変数である `a` は `main.c` で外部変数宣言されているので、静的記憶領域に確保されています。
⑦の `n` は引数を格納するための内部変数であるため、スタック領域に確保されます。

Lesson3 変数のスコープ

問題 1	①2 ②2 ③3 ④12
問題 2	①5 ②8 ③40
問題 3	①イ ②エ ③オ