

第11回

プログラミング及び実習 II

1

第10回課題の解答例

2

ex10-1：なにも値を返さない関数

```
#include <stdio.h>

void pr();

int main() {
    pr();
    return 0;
}

void pr() {
    printf("なにも値を返さない関数 pr() を呼び出しました\n");
    return;
}
```

3

ex10-2：返り値のある関数

```
#include <stdio.h>

// 関数 sc() のプロトタイプ宣言
int sc();

int main() {
    printf("入力された値は %d\n", sc());
    return 0;
}

int sc() {
    int a;
    scanf("%d", &a);
    return a;
}
```

4

ex10-3 : 2つの関数定義

```
#include <stdio.h>

void pr2();
int sc();

int main() {
    int a = sc();
    pr2();
    printf("%d\n", a);
    return 0;
}

void pr2() {
    printf("入力された値は ");
    return;
}

int sc() {
    int a;
    scanf("%d", &a);
    return a;
}
```

5

ex10-4 : 黄金比

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

void pexit(void);
double gratio(void);

int main(void) {
    double a;

    while (1) {
        scanf("%lf", &a);
        if (a <= 0) pexit();

        printf("%.4f\n", a*gratio());
    }
    return 1;
}

void pexit(void) {
    printf("終了します。 \n");
    exit(0);
}

double gratio(void) {
    return (1+sqrt(5))/2;
}
```

6

ex10-5 : 2科目の統計量

```
#include <stdio.h>

int main() {
    int n;
    do {
        scanf("%d", &n);
    } while(n<=0);

    int a[n], b[n];
    int amax, bmax, tmax;
    int amin, bmin, tmin;
    int asum, bsum, tsum;
    asum = bsum = tsum = 0;

    for (int i=0; i<n; i++) {
        scanf("%d%d", &a[i], &b[i]);

        int total = a[i]+b[i];
        asum += a[i], bsum += b[i], tsum += total;

        if (amax<a[i] || i==0) amax = a[i];
        if (amin>a[i] || i==0) amin = a[i];
        if (bmax<b[i] || i==0) bmax = b[i];
        if (bmin>b[i] || i==0) bmin = b[i];
        if (tmax<total || i==0) tmax = total;
        if (tmin>total || i==0) tmin = total;
    }

    printf("%d %d %.2f\n", amax, amin, (double)asum/n);
    printf("%d %d %.2f\n", bmax, bmin, (double)bsum/n);
    printf("%d %d %.2f\n", tmax, tmin, (double)tsum/n);

    return 0;
}
```

7

ex10-6 : 全角のアスタリスク

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

int main() {
    int i, j, n;
    scanf("%d", &n);

    if (n < 1) exit(1);

    for (i=0; i<2*n+1; i++) {
        for (j=0; j<2*n+1; j++)
            if (i <= -j+n || i >= j+n || i <= j-n || i >= -j+3*n)
                printf("*");
            else
                printf(" ");
        printf("\n");
    }
    return 0;
}
```

8

ex10-7 : 全角のアスタリスク2

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

int main() {
    int i, j, n;
    scanf("%d", &n);

    if (n < 1) exit(1);

    for (i=0; i<2*n+1; i++) {
        for (j=0; j<2*n+1; j++)
            if ((abs(n-i)+abs(n-j))%3 == 0)
                printf("*");
            else
                printf(" ");
            printf("\n");
        }
    return 0;
}
```

9

C言語の関数

数学の関数

引数のないプログラミング言語の関数 `int func()`

`func()` `int v`

引数があるプログラミング言語の関数 `int farg(a)`

`int a` `farg(a)` `int v`

11

第11回の内容

C言語の関数++

引数のある関数

配列変数を引数とする関数

10

第11回の内容

C言語の関数++

引数のある関数

配列変数を引数とする関数

12

引数がなく返り値のある関数の宣言と定義

```
#include <stdio.h>
int ten(void);
int main(void) {
    printf("%d\n", ten());
    return 0;
}
int ten(void) {
    return 10;
}
```

引数型を void として「引数がないこと」
を明示的に表す (省略可能)

int 型の関数 ten() のプロトタイプ宣言

関数 ten() の呼び出しと返り値の利用

関数 ten() の定義

関数を評価する (呼び出す) と定義された型の値を
(返り値として) もつ main() 0

返り値の値は、関数定義の return 文で与えられる 10

関数の型と return 文で返す式 (値) の型は一致 ten()

13

引数があり返り値がない関数の宣言と定義

```
#include <stdio.h>
void prsq(int);
int main(void) {
    prsq(3);
    return 0;
}
void prsq(int a) {
    printf("%d\n", a*a);
}
```

void 型の関数 prsq() のプロトタイプ宣言

void prsq(int b); としても間違いではない

関数 prsq() の呼び出し

関数 prsq() の定義

関数に渡される値 (や変数) を引数と呼ぶ: prsq(3) main() 0

関数 prsq() は整数型の引数を1つもつ: void prsq(int) 3

関数に引数として渡された値は、その関数の実行時に
(引数として) 定義された変数に代入される: int a = 3; prsq()

14

引数があり返り値もある関数の宣言と定義

```
#include <stdio.h>
int square(int);
// int square(int b);
int main(void) {
    printf("%d\n", square(3));
    return 0;
}
int square(int a) {
    return a*a;
}
```

int 型の関数 square() のプロトタイプ宣言

関数 square() の呼び出し

関数 square() の定義

関数に渡される値 (や変数) を引数と呼ぶ: square(3) main() 0

関数 square() は整数型の引数を1つもつ: int square(int) 3 9

関数に引数として渡された値は、その関数の実行時に
(引数として) 定義された変数に代入される: int a = 3; square()

15

複数の引数がある関数の宣言と定義

```
#include <stdio.h>
int add(int, int);
// int add(int a, int b);
int main(void) {
    printf("%d\n", add(3, 5));
    return 0;
}
int add(int a, int b) {
    return a+b;
}
```

int 型の関数 add() のプロトタイプ宣言

関数 add() の呼び出し

関数 add() の定義

関数 add() は整数型の引数を2つもつ:
int add(int, int) main() 0

関数に引数として渡された値は、その関数の実行時に (引
数として) 定義された変数に代入される: int a = 3, b = 5; 3,5 8

呼び出し時に引数の数が異なるとエラー: add(3) add()

16

第11回の内容

C言語の関数++

引数のある関数

配列変数を引数とする関数

17

配列変数のサイズは関数で直接渡せない

関数呼び出しで `sum(a[10])`; としても「大きさ10の配列を渡す」ことにならない

`sum(a, n)` のように配列変数名と配列変数の大きさ (サイズ) を引数とする

あるいは、関数にサイズが伝われば良いので `#define` で定義することもできる

内実に興味があればポインタ変数について学んでください (後継科目でやります)

```
#include <stdio.h>
#define N 5                #define で定数定義して関数にサイズを与える

int sum(int[]);
// int sum (int[], int);

int main(void) {
    int a[N] = {1,2,3,4,5};
    sum(a);
    return 0;
}

int sum(int a[]) {
    int sum = 0;
    for (int i=0; i<N; i++) {...} // 配列 a[] の N 個の要素を sum に加算
    return sum;
}
```

19

配列変数を引数とする関数の宣言と定義

```
#include <stdio.h>

int sum(int[], int);      int 型の関数 sum() のプロトタイプ宣言
// int sum (int x[], int n); 第1引数は int型の配列変数であることを宣言
                             第2引数では int型で配列サイズを渡す

int main(void) {
    int a[5] = {1,2,3,4,5};
    sum(a, 5);            sum() に配列変数の名前とサイズを渡す
    return 0;
}

int sum(int a[], int n) {      関数 sum() の定義
    int sum = 0;
    // 配列 a[] の n 個の要素を sum に加算
    return sum;
}
```

配列変数の引数は、配列を表す [] 付きで宣言・定義

配列変数の大きさ (サイズ) は直接渡せないため、他の引	main()	0
数として関数に渡す (sum() 関数の第2引数 n)	{1,2,3,4,5}	15

関数の呼び出し時は、配列変数の名前のみで [] が付かないことに注意 (`sum(a, 5);`)

	sum()
--	-------

18

引数で可変長配列を使う : C99規格

C99規格のC言語では、配列のサイズを実行時に決めることができる (第5回)

```
int n = 10;
int a[n];           // int a[10]
```

この可変長配列は関数の引数としても利用可能

```
int sum(int size, int a[size]) { ... } // 関数定義
sum(10, a);                             // 呼び出し
```

可変長配列は注意して利用しましょう

実行時に `n = 1000000000` や `n = -1` となる危険

C99規格でしか使えない (新しい規格ではオプション扱い)

20