

プログラミング演習

バージョン1

担当教員：綴木 馴

●プログラムの決まりについて学ぶ

おすすめする参考書

ザ・C 戸川隼人 サイエンス社

- ・本日の予定
 1. 授業の説明.
 2. コンパイラーのインストール.

●プログラムの決まりについて学ぶ, P31

```
/* The most ..... in C */  
/* hello.c */  
#include <stdio.h>  
main()  
{  
    printf("hello,world¥n");  
}
```

●プログラムの決まり(コメント)

```
/* The most ..... in C */  
/* hello.c */  
#include <stdio.h>  
main()  
{  
    printf("hello,world¥n");  
}
```

コメントは

/* */

でくる

●プログラムの決まり(コメント)

```
/* The most ..... in C */  
/* hello.c */  
#include <stdio.h>  
main()  
{  
    printf("hello,world¥n");  
}
```

まずは**決まり文句**と思ってください

●プログラムの決まり(関数)

```
/* The most ..... in C */  
/* hello.c */  
#include <stdio.h>  
main()  
{  
    printf("hello,world¥n");  
}
```

出力させたい内容を
「"」
でくる

関数は
必ず「;」で
終わる

画面に出力する
出力関数と呼ぶ

¥nは改行を
意味する

●プログラミングの決まり(行の概念は無い)

```
/* The most ..... in C */  
/* hello.c */  
#include <stdio.h>  
main()  
{  
    printf("hello,world¥n");  
}
```



```
/* The most ..... in C */  
/* hello.c */  
#include <stdio.h>  
main(){printf("hello,world¥n");}
```

課題

- P35, 4. 3, 4. 4, 4. 5

●今日のプログラム(入出力の関数)P36

```
#include <stdio.h>
main()
{
    int data;
    printf("Input an integer :");
    scanf("%d",&data);
    printf("The interger is  %d¥n",data);
}
```

●変数の取り扱い

```
#include <stdio.h>
main()
{
    int data;
    printf("Input an integer :");
    scanf("%d",&data);
    printf("The interger is  %d¥n",data);
}
```

dataという**整数型**の変数
を宣言する
(変数を使えるようにする)

整数型	int	1,2,100,-4 など
文字型	char	a,b,c など
実数型	float	1.2, 100, 34.1 など

●出力関数の取り扱い

```
#include <stdio.h>
main()
{
    int data;
    printf("Input an integer :");
    scanf("%d",&data);
    printf("The interger is  %d¥n",data);
}
```

&を忘れるな
(今は決まり文句と
思っておく)

入力関数と呼ぶ

キーボードから%dに入力された整数をdataに保存する

整数型	%d
文字型	%c
実数型	%f

10進数	%d
8進数	%o
16進数	%x

●出力関数の取り扱い

```
#include <stdio.h>
main()
{
    int data;
    printf("Input an integer :");
    scanf("%d",&data);
    printf("The interger is %d¥n",data);
}
```

dataの値を画面に出力する

課題

- P43 5.1, 5.2, 5.3

●第6章 簡単な計算p44

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r
    printf("The area is %f¥n",s);
}
```

新しいポイント

double

%lf

* (掛け算の記号つまり×)

●第6章 簡単な計算

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r
    printf("The area is %f¥n",s);
}
```

新しいポイント

double について

●倍精度実数型

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r
    printf("The area is %f¥n",s);
}
```

floatよりも
倍の精度
(例えば: **小数点の
桁数が多い**)
の値が使える

知識の追加

今までの知識:

}	整数型	int	1,2,100,-4 など
	文字型	char	a,b,c など
	実数型	float	1.2, 100, 34.1 など

新しい知識: 倍精度実数型 **double** 1.2, 100, 34.1 など

●第6章 簡単な計算

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r
    printf("The area is %f¥n",s);
}
```

新しいポイント

double

`%lf` について

●倍精度実数型変数への入力

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r
    printf("The area is %f¥n",s);
}
```

知識の追加

今までの知識:

整数型	%d
文字型	%c
実数型	%f

10進数	%d
8進数	%o
16進数	%x

新しい知識:

倍精度実数型	%lf
--------	-----

●倍精度実数型変数への入力

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r;
    printf("The area is %f¥n",s);
}
```

&を忘れるな
(今は決まり文句と
思っておく)

知識の追加

キーボードから%lfに入力された倍精度実数をrに保存する

新しい知識:

倍精度実数型

%lf

● 四則演算

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r
    printf("The area is %f¥n",s);
}
```

知識の追加

*	掛け算
/	割り算
+	足し算
-	引き算
%	わり算の余り

●今日の注意

```
#include <stdio.h>
main()
{
    double r, s;
    printf("radius = ");
    scanf("%lf",&r);
    s = 3.14 * r * r
    printf("The area is %f¥n",s);
}
```

sは**倍精度実数型**で宣言されているが普通の実数型で出力されている

課題

- P54 6.1, 6.2, 6.3

●今日の目的: 条件分岐 (if 文) P67

```
#include <stdio.h>
#include <math.h>
main(){
    double  a, b, c, d, rd, x1, x2;
    printf(" a= "); scanf("lf",&a);
    printf(" b= "); scanf("lf",&b);
    printf(" c= "); scanf("lf",&c);
    d = b * b - 4 * a * c;
    if(d >= 0)
    {
        rd = sqrt(d);
        x1 = (-b - rd) / (2 * a);
        x2 = (-b + rd) / (2 * a);
        printf("Solution_1 = %f¥n", x1);
        printf("Solution_2 = %f¥n", x2);
    }
    else printf("No real solution¥n");
}
```

●今日の目的: 条件分岐 (if 文)

```
#include <stdio.h>
#include <math.h>
main(){
    double  a, b, c, d, rd, x1, x2;
    printf(" a= "); scanf("lf",&a);
    printf(" b= "); scanf("lf",&b);
    printf(" c= "); scanf("lf",&c);
    d = b * b - 4 * a * c;
    if(d >= 0)
    {
        rd = sqrt(d);
        x1 = (-b - rd) / (2 * a);
        x2 = (-b + rd) / (2 * a);
        printf("Solution_1 = %f¥n", x1);
        printf("Solution_2 = %f¥n", x2);
    }
    else printf("No real solution¥n");
}
```

新しい箇所

#include <math.h>

sqrt(d)

if

else

●平方根を取る関数: sqrt()

```
#include <stdio.h>
#include <math.h>
main(){
    double  a, b, c, d, rd, x1, x2;
    printf(" a= "); scanf("lf",&a);
    printf(" b= "); scanf("lf",&b);
    printf(" c= "); scanf("lf",&c);
    d = b * b - 4 * a * c;
    if(d >= 0)
    {
        rd = sqrt(d);
        x1 = (-b - rd) / (2 * a);
        x2 = (-b + rd) / (2 * a);
        printf("Solution_1 = %f¥n", x1);
        printf("Solution_2 = %f¥n", x2);
    }
    else printf("No real solution¥n");
}
```

sqrt を使えるようにする
ための決まり文句

カッコ内()の
ルート(√)
を計算する

●今日の目的: 条件分岐 (if 文)

```
#include <stdio.h>
#include <math.h>
main(){
    double  a, b, c, d, rd, x1, x2;
    printf(" a= "); scanf("lf",&a);
    printf(" b= "); scanf("lf",&b);
    printf(" c= "); scanf("lf",&c);
    d = b * b - 4 * a * c;
    if(d >= 0)
    {
        rd = sqrt(d);
        x1 = (-b - rd) / (2 * a);
        x2 = (-b + rd) / (2 * a);
        printf("Solution_1 = %f¥n", x1);
        printf("Solution_2 = %f¥n", x2);
    }
    else printf("No real solution¥n");
}
```

新しい箇所

if(条件)

(条件)を満たすならば
すぐ下の{ }内を実行

else

(条件)を満たさないならば
すぐ下の{ }内を実行

1行の時のみ{ }
内を実行できる

●今日の目的: 条件分岐 (if 文)

```
#include <stdio.h>
#include <math.h>
main(){
    double  a, b, c, d, rd, x1, x2;
    printf(" a= "); scanf("lf",&a);
    printf(" b= "); scanf("lf",&b);
    printf(" c= "); scanf("lf",&c);
    d = b * b - 4 * a * c;
    if(d >= 0)
    {
        rd = sqrt(d);
        x1 = (-b - rd) / (2 * a);
        x2 = (-b + rd) / (2 * a);
        printf("Solution_1 = %f¥n", x1);
        printf("Solution_2 = %f¥n", x2);
    }
    else printf("No real solution¥n");
}
```

新しい箇所

if(条件)

条件式の定義

$a==b \rightarrow a=b$ のとき

$a!=b \rightarrow a \neq b$ のとき

$a > b \rightarrow a > b$ のとき

$a < b \rightarrow a < b$ のとき

$a \geq b \rightarrow a \geq b$ のとき

$a \leq b \rightarrow a \leq b$ のとき

課題

- P72 8.1, 8.2, 8.3

●今日の目的: 繰り返し (**while** 文, P81)

```
#include <stdio.h>
#include <math.h>
#define EPS 1e-5
main()
{
    double a, old_x, new_x;
    int i;

    printf(" a= "); scanf("%lf", &a);

    new_x = a; i = 0;
    while(i < 100)
    {
        old_x = new_x;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f¥n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
        i++;
    }
}
```

新しい箇所

#define

while

i++;

break;

fabs

●今日の目的: 繰り返し(**while** 文, P81)

```
#include <stdio.h>
#include <math.h>
#define EPS 1e-5
main()
{
    double a, old_x, new_x;
    int i;

    printf(" a= "); scanf("%lf", &a);

    new_x = a; i = 0;
    while(i < 100)
    {
        old_x = new_x;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f¥n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
        i++;
    }
}
```

EPSを
 10^{-5}
と定義する

●今日の目的: 繰り返し (**while** 文, P81)

```
#include <stdio.h>
#include <math.h>
#define EPS 1e-5
main()
{
    double a, old_x, new_x;
    int i;

    printf(" a= "); scanf("%lf, &a);

    new_x = a; i = 0;
    while(i < 100)
    {
        old_x = new_x;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f¥n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
        i++;
    }
}
```

while(条件)

(条件)が成立している場合は
{ }内を繰り返す

●今日の目的: 繰り返し (**while** 文, P81)

```
#include <stdio.h>
#include <math.h>
#define EPS 1e-5
main()
{
    double a, old_x, new_x;
    int i;

    printf(" a= "); scanf("%lf", &a);

    new_x = a; i = 0;
    while(i < 100)
    {
        old_x = new_x;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f¥n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
        i++;
    }
}
```

fabsが使えるようにする

while文から抜け出す.

()内の絶対値をとる

●今日の目的: 繰り返し (**while** 文, P81)

```
#include <stdio.h>
#include <math.h>
#define EPS 1e-5
main()
{
    double a, old_x, new_x;
    int i;

    printf(" a= "); scanf("%lf", &a);

    new_x = a; i = 0;
    while(i < 100)
    {
        old_x = new_x;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f¥n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
        i++;
    }
}
```

i++;

iに1を足す
i=i+1;
または
++i;
と書いても良い

●今日の目的: 繰り返し2 (**while** 文, P82)

```
#include <stdio.h>
main()
{
    double ave, sum=0;    int    data, num=0;
    printf("data = ");
    while(1)
    {
        scanf("%d", &data);
        if(data == 12345)break;
        sum += data;
        num++;
        printf("data = ") ;
    }
    if(num == 0)
        printf("can't calculate the average¥n");
    else
    {
        ave = sum / num;
        printf("the average of %d data is %f¥n", num , ave);
    }
}
```

新しい箇所

+=

課題

- P84 9.1, 9.2, 9.3

●今日の目的: 繰り返し2 (**while** 文, P82)

```
#include <stdio.h>
main()
{
    double  ave, sum=0;    int    data, num=0;
    printf("data = ");
    while(1)
    {
        scanf("%d", &data);
        if(data == 12345)break;
        sum += data;
        num++;
        printf("data = ") ;
    }
    if(num == 0)
        printf("can't calculate the average¥n");
    else
    {
        ave = sum / num;
        printf("the average of %d data is %f¥n", num , ave);
    }
}
```

新しい箇所

+=

sum += data;
は
sum = sum +data;
と同じ.

● 課題

P84, 9.4, 9.5, 9.6

●今日の目的: 繰り返し (do, while 文, P97)

```
#include <stdio.h>
#include <time.h>
main()
{
    int            d, x;
    unsigned       seed;
    printf("Input an integer to seed =");
    scanf("%d", &seed); srand(seed);
    x = rand();
    x %= 100;
    printf("data = ");
    do
    {
        printf("Hit my number(0-99)");
        scanf("%d", &d);
        if(x > d) printf("greater than %d ¥n", d);
        else if(x < d) printf("less than %d ¥n", d);
    }
    while(x != d);
    printf("Congratulations !¥n");
}
}
```

新しい箇所

unsigned

srand()

rand()

do, while

●今日の目的: 繰り返し (do, while 文, P97)

```
#include <stdio.h>
#include <time.h>
main()
{
    int            d, x;
    unsigned       seed;
    printf("Input an integer to seed =");
    scanf("%d", &seed); srand(seed);
    x = rand();
    x %= 100;
    do{
        printf("Hit my number(0-99)");
        scanf("%d", &d);
        if(x > d) printf("greater than %d ¥n", d);
        else if(x < d) printf("less than %d ¥n", d);
    }
    while(x != d);

    printf("Congratulations !¥n");
}
}
```

unsigned

符号なしの
正数つまり
正の整数のみ
の場合に使用

●今日の目的: 繰り返し (do, while 文, P97)

```
#include <stdio.h>
#include <time.h>
main()
{
    int            d, x;
    unsigned       seed;
    printf("Input an integer to seed =");
    scanf("%d", &seed); srand(seed);
    x = rand();
    x %= 100;
    do{
        printf("Hit my number(0-99)");
        scanf("%d", &d);
        if(x > d) printf("greater than %d ¥n", d);
        else if(x < d) printf("less than %d ¥n", d);
    }
    while(x != d);

    printf("Congratulations !¥n");
}
}
```

srand()

乱数の種を
与える

rand()

乱数を発生
させる

●今日の目的: 繰り返し (do, while 文, P97)

```
#include <stdio.h>
#include <time.h>
main()
{
    int            d, x;
    unsigned       seed;
    printf("Input an integer to seed =");
    scanf("%d", &seed); srand(seed);
    x = rand();
    x %= 100;
    do{
        printf("Hit my number(0-99)");
        scanf("%d", &d);
        if(x > d) printf("greater than %d ¥n", d);
        else if(x < d) printf("less than %d ¥n", d);
    }
    while(x != d);

    printf("Congratulations !¥n");
}
}
```

do, while

while(条件)
条件を
満たしている間
{ }内を繰り返す.

● 課題

P98, 10.2, 10.3, 10.4

● for 文 (P89)

新しいポイント

```
#include <stdio.h>
#include <math.h>
#define EPS 1e-5
main()
{
    double a, old_x, new_x;
    int i;
    printf(" a ="); scanf("%lf", &a);

    new_x = a;
    for(i = 0; i < 100; i++)
    {
        old_x = new_x ;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f\n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
    }
}
```

for(初期設定; 反復条件; 変更処理)

● for 文 (P89)

新しいポイント

```
#include <stdio.h>
#include<math.h>
#define      EPS 1e-5
main()
{
    double a, old_x, new_x;
    int    i;
    printf(" a =");  scanf("%lf", &a);

    new_x = a;
    for(i = 0; i < 100; i++)
    {
        old_x = new_x ;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f\n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
    }
}
```

for(初期設定; 反復条件; 変更処理)

● for 文 (P89)

新しいポイント

```
#include <stdio.h>
#include<math.h>
#define      EPS 1e-5
main()
{
```

for(初期設定; 反復条件; 変更処理)

```
    double a, old_x, new_x;
    int     i;
    printf(" a =");  scanf("%lf", &a);
```

```
    new_x = a;
    for(i = 0; i < 100; i++)
    {
```

i=0と初期設定する

```
        old_x = new_x ;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f\n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
    }
```

```
}
}
```

● for 文 (P89)

新しいポイント

```
#include <stdio.h>
#include<math.h>
#define      EPS 1e-5
main()
{
```

for(初期設定; 反復条件; 変更処理)

```
    double a, old_x, new_x;
    int    i;
    printf(" a =");  scanf("%lf", &a);
```

iが100より
小さい間{ }内を
繰り返す

```
    new_x = a;
    for(i = 0; i < 100; i++)
    {
        old_x = new_x ;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f\n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
    }
}
```

● for 文 (P89)

新しいポイント

```
#include <stdio.h>
#include<math.h>
#define      EPS 1e-5
main()
{
```

for(初期設定; 反復条件; 変更処理)

```
    double a, old_x, new_x;
    int    i;
    printf(" a =");  scanf("%lf", &a);
```

```
    new_x = a;
    for(i = 0; i < 100; i++)
    {
```

iに1を加える

```
        old_x = new_x ;
        new_x = (old_x + a / old_x) / 2;
        printf("x = %f\n", new_x);
        if(fabs(new_x - old_x)/old_x < EPS) break;
    }
```

```
}
}
```

●課題

1. 1から1000までの和をfor文を使って求めよ.
2. P98, 10.1(cf p82), 10.5

● 文字型の配列(文字列)

```
#include <stdio.h>
main()
{
char  a[100];
int    i = 0;
print("何か文字列を入力して下さい. ");
scanf("%s", a);
while(a[i] != 0){
    printf("a[%d] = %c¥n", i, a[i]);
    i++;
}
}
```

新しいポイント

a[100]

%s

文字列を入力し、先頭から一字ずつ取り出して表示するプログラム

● 文字型の配列(文字列)

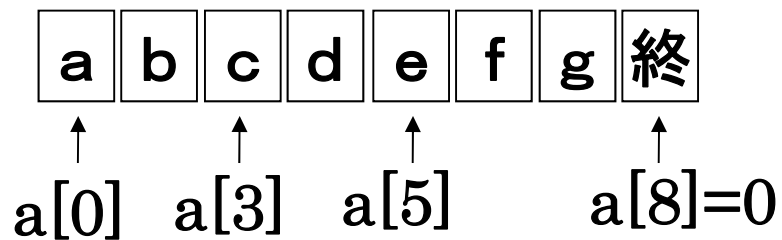
```
#include <stdio.h>
main()
{
char  a[100];
int    i = 0;
printf("何か文字列を入力して下さい. ");
scanf("%s", a);
while(a[i] != 0){
    printf("a[%d] = %c¥n", i, a[i]);
    i++;
}
}
```

新しいポイント

a[100]

100個の文字型
配列を確保

例: abcdefgと入力した場合



● 文字型の配列(文字列)

```
#include <stdio.h>
main()
{
char  a[100];
int    i = 0;
print("何か文字列を入力して下さい. ");
scanf("%s", a);
while(a[i] != 0){
    printf("a[%d] = %c¥n", i, a[i]);
    i++;
}
}
```

新しいポイント

文字列型 %s

文字型配列
a[100]に
文字列をキーボ
ードから入力

復習:

整数型	%d
文字型	%c
実数型	%f

10進数	%d
8進数	%o
16進数	%x

● 文字型の配列(文字列)の課題

1. 例題を do while 文で書き直せ.
2. 例題を for 文で書き直せ.
3. abcdeと入力し,a[0]~a[6]を整数で表示してみよ.
4. 入力された文字列を一発表示せよ.
ヒント:printf("%s", a);

●文字列のコピー

```
main()
{
int    i;
char   a[100], b[100];
for(i = 0; i<100; i++)
    b[i] = 0;
printf("文字列を入力して下さい");
scanf("%s", a);
for(i = 0; a[i] != 0; i++)
    b[i] = a[i];
printf("文字列 b を表示します %s¥n", b);
}
```

新しいポイント

特になし

●文字列のコピー(課題)

1. 例題を do while 文で書き直せ.
2. 例題を while 文で書き直せ.
3. 10個の配列を用意し, 0~99までの整数を
10個ランダムに生成して配列に代入し表示せよ.

●2次元配列

```
main()
{
int    i;
char   a[2][100];
for(i = 0; i<100; i++)
    a[1][i] = 0;
printf("文字列を入力して下さい");
scanf("%s", a[0]);
for(i = 0; a[0][i] != 0; i++)
    a[1][i] = a[0][i];
printf("文字列 a[1] を表示します %s¥n", a[1]);
}
```

●2次元配列

```
main()
{
int    i;
char   a[2][100];
for(i = 0; i<100; i++)
    a[1][i] = 0;
printf("文字列を入力して下さい");
scanf("%s", a[0]);
for(i = 0; a[0][i] != 0; i++)
    a[1][i] = a[0][i];
printf("文字列 a[1] を表示します %s¥n", a[1]);
}
```

新しいポイント

a[2][100];

●2次元配列

```
main()
{
int    i;
char   a[2][100];
for(i = 0; i<100; i++)
    a[1][i] = 0;
printf("文字列を入力して下さい");
scanf("%s", a[0]);
for(i = 0; a[0][i] != 0; i++)
    a[1][i] = a[0][i];
printf("文字列 a[1] を表示します %s¥n", a[1]);
}
```

新しいポイント

a[2][100];

a[0][0], a[0][1], a[0][2], , a[0][98], a[0][99]

a[1][0], a[1][1], a[1][2], , a[1][98], a[1][99]

●2次元配列(課題)

1. 例題を do while 文で書き直せ.
2. 例題を while 文で書き直せ.
3. 10個の配列を用意し, 0~99までの整数を10個ランダムに生成してその配列に代入し, その中から最大値を求めよ.

●2次元配列(整数型)

```
main()
{
int    i;
int    a[2][100];
for(i = 0; i<100; i++)
{
    printf("整数を入力せよ(0で終了)");
    scanf("%d", &a[0][i]);
    if(a[0][i]==0) break;
}
for(i = 0; a[0][i] != 0; i++)
{
    a[1][i] = a[0][i];
    printf("整数 a[1][%d] %d¥n", i, a[1][i]);
}
}
```

●2次元配列(課題)

1. 0~99までの整数を10個ランダムに生成し、大きい順に並び替えるプログラムを作成せよ.

