

プログラミング基礎演習A



数学関数
乱数

キャスト演算子

(型名)変数名

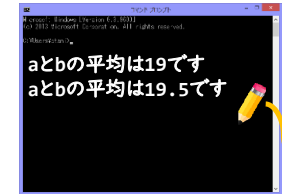
◆ 変数の値を型名で指定された型の値に変換する

```
int a, b, ave1;
double ave2;

a = 16;
b = 23;
ave1 = (a + b) / 2;
printf("aとbの平均は%dです\n", ave1);
ave2 = (double)(a + b) / 2;
printf("aとbの平均は%fです\n", ave2);
```

商は整数

doubleに変換される



数学関数

- ◆ ヘッダファイル`math.h`を読み込むことで使用可能
- ◆ 例外を除き, 引数と戻り値の型は`double`
- ◆ 角度は弧度法(ラジアン) $\pi \text{rad} = 180^\circ$

◆ よく使う数学関数

- | | |
|----------------------------------|--|
| ◆ <code>sqrt(x)</code> : xの平方根 | ◆ <code>sin(x)</code> : xの正弦値 |
| ◆ <code>pow(x, y)</code> : xのy乗 | ◆ <code>cos(x)</code> : xの余弦値 |
| ◆ <code>exp(x)</code> : eのx乗 | ◆ <code>tan(x)</code> : xの正接値 |
| ◆ <code>log(x)</code> : xの自然対数 | ◆ <code>ceil(x)</code> : x以上の整数のうち, 最小の整数 |
| ◆ <code>log10(x)</code> : xの常用対数 | ◆ <code>floor(x)</code> : x以下の整数のうち, 最大の整数 |
| ◆ <code>fabs(x)</code> : xの絶対値 | |

プログラム例

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

int main()
{
    double area, side, rad, sine;

    printf("正方形の面積:");
    scanf("%lf", &area);
    side = sqrt(area);
    printf("正方形の一辺の長さは%fです.\n", side);

    printf("角度(rad):");
    scanf("%lf", &rad);
    sine = sin(rad);
    printf("sin(%f)=%f\n", rad, sine);
}
```

平方根の計算

正弦の計算

乱数列

- ◆ 規則性がなく、次に現れる数値が予測不能な数値の列
例) サイコロ, コイン投げ, くじ引き etc.

これが
乱数

- ◆ 決められた動作しかないコンピュータでは乱数列の生成は不可能
→ 疑似乱数列を使用

漸化式に初期シードを与えて生成



いろいろな
疑似乱数列の
生成方法が
あるよ!

5

乱数関数

`srand(初期シード)`

乱数の種とも呼ぶ
型は符号なし整数(unsigned)

- ◆ 初期シードに基づき, 0以上RAND_MAX以下の整数の疑似乱数列を生成する
- ◆ 乱数を使うときは最初に一度だけ必ず呼び出す
- ◆ ヘッダファイル`stdlib.h`が必要

呼び出さないとデフォルトの疑似
乱数列が使用される
→ 実行を繰り返しても発生する
乱数は変わらない

`rand()`

- ◆ 呼び出されるたびに, 生成した疑似乱数列の値を1つずつ返す
- ◆ ヘッダファイル`stdlib.h`が必要

6

乱数関数の活用

- ◆ 0~RAND_MAXの整数以外の乱数を発生させたいときはどうする?
→ 剰余や商を活用する
 - ◆ 0または1の乱数 : `rand() % 2`
 - ◆ 0以上N以下の整数の乱数 : `rand() % (N + 1)`
 - ◆ M以上N以下の整数の乱数 : `rand() % (N - M + 1) + M`
 - ◆ 0以上1以下の実数の乱数 : `(double)rand() / RAND_MAX`
- ◆ 実行のたびに異なる乱数を発生させるにはどうする?
→ 実行のたびに異なる初期シードを渡す
→ 実行日時により決まる値を初期シードにする!



わる数と
あまりの関係を
考えてみよう!

7

time関数

`time(エポック秒を格納する変数)`

1970年1月1日からの経過秒
UNIX時間とも呼ぶ

- ◆ エポック秒を取得し, 引数として渡された変数に格納するとともに, 戻り値としても返す
- ◆ 引数 : エポック秒を格納する変数へのポインタ
格納する必要がないときはNULLとする
- ◆ 戻り値 : エポック秒(time_t型)
- ◆ ヘッダファイル`time.h`が必要

初期シードは
`srand((unsigned)time(NULL))`
と設定すればよい!

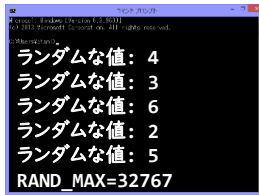
8

プログラム例

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

```
int main()
{
    int i, dice;

    srand((unsigned)time(NULL));
    for(i = 0 ; i < 5 ; i++) {
        dice = rand() % 6 + 1;
        printf("ランダムな値: %d\n", dice);
    }
    printf("RAND_MAX=%d\n", RAND_MAX);
}
```



```
ランダムな値: 4
ランダムな値: 3
ランダムな値: 6
ランダムな値: 2
ランダムな値: 5
RAND_MAX=32767
```

乱数なので実行するたびに
値は変わる
RAND_MAXは不変

1以上6以下の整数の乱数
= 0以上5以下の整数の乱数 + 1
= rand() % 6 + 1

RAND_MAXの値を確認

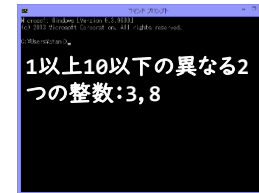
9

演習

1以上10以下の整数のうち、2つの異なる整数をランダムに選んで表示するプログラムを作成せよ。

プログラム名はe7とすること。

<実行例>



```
1以上10以下の異なる2
つの整数:3,8
```

10

演習 ～手順～

1. 中身が空のメイン関数を書く
2. 必要なヘッダファイルは?
 - 入力や出力には**stdio.h**が必要
 - 乱数関数の使用には**stdlib.h**が必要
 - time関数の使用には**time.h**が必要
 - #includeの行を書く
3. 必要な変数は?
 - 2つの整数を代入する変数
 - 型と変数名を決めて変数を宣言する

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

int main()
{
    int r1, r2;
}
```

乱数を使うとき
2つのヘッダファイル
は必須



11

演習 ～手順～

4. 処理の順番は?
 - 乱数の種を設定する (**srand** + **time**)
 - 1つ目の乱数を発生させる (**rand**)
 - 2つ目の乱数が1つ目の乱数と異なる値になるまで以下を繰り返す (**do-while** 文)
 - 2つ目の乱数を発生させる (**rand**)
 - 2つの乱数を表示する (**printf**)

1以上10以下の整数の乱数
= 0以上9以下の整数の乱数 + 1
= rand() % 10 + 1

異なる整数が
出るまで
繰り返すんだね



12