

情報研究会CACTUS

第2回 講義資料

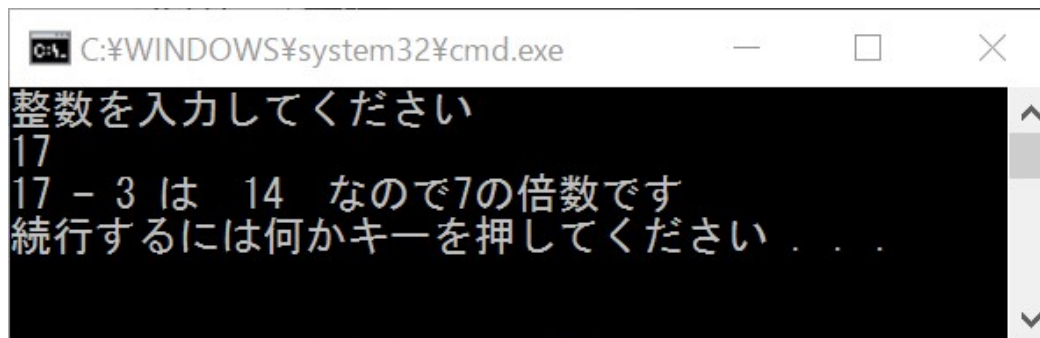
今回の内容

- ・ 前回のおさらい問題
- ・ switch文
- ・ while文
- ・ for文
- ・ 数学関数

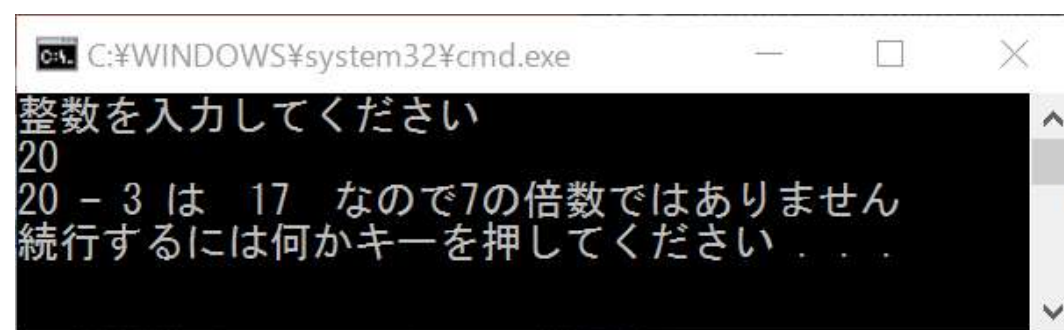
前回のおさらい問題

入力された整数から3を引いた値が、
7の倍数になるかどうかを判定するプログラム
(制限時間 30分)

例) 17 と 20 の場合



```
C:\¥WINDOWS¥system32¥cmd.exe
整数を入力してください
17
17 - 3 は 14 なので7の倍数です
続行するには何かキーを押してください . . .
```



```
C:\¥WINDOWS¥system32¥cmd.exe
整数を入力してください
20
20 - 3 は 17 なので7の倍数ではありません
続行するには何かキーを押してください . . .
```

サンプルコード1

入力された整数を
3で割った余りを
判定するプログラム

```
1  #include<stdio.h>
2
3  int main(void)
4  {
5      int n;
6
7      printf("整数を入力してください:");
8      scanf("%d", &n);
9
10     //switch文を使う場合
11     switch (n % 3)
12     {
13     case 1:
14         printf("%dを3で割った余りは1です\n", n);
15         break;
16     case 2:
17         printf("%dを3で割った余りは2です\n", n);
18         break;
19     default:
20         printf("%dは3の倍数です\n", n);
21     }
22     /*switch文を使わない場合
23     printf("%dを3で割った余りは%dです\n", n, n%3);*/
24
25     return 0;
26 }
```

switch文

式の値が設定した定数になったとき、
処理を開始する。

式と定数の値に
実数は使用できない。

```
switch( 式 )  
{  
    case 定数a :    処理1  
  
    case 定数b :    処理2  
  
    default : 処理3  
}
```

switch文

上から順番に判定していく。

```
switch(x){  
    case 0:  
    case 1:  
        printf(" 0 か 1 だよ" ); break;  
    case2: printf(" 2 だよ" ); break;  
    default: printf(" 3以上だよ" );  
}
```

二個以上の場合は
並列してかく

caseの横に条件を
書くこともできる

「それ以外」の時は
defaultと書く

break とは

switch文や繰り返しの処理を途中で終了させる。

これをしないと、それ以降に書かれている処理をすべて行われてしまう。

サンプルコード1のコードのbreakを消して確認してみよう。

コメントアウト

コードに関係ない部分（読んだ人に対する説明など）は`/**/`で囲う、または`//`と書く。

例)

`/*この文は
表示されないよ！*/`

`//この文も表示されないよ！`

二行以上書くときは`/**/`,
一行の時は`//`を使う

確認問題1

サンプルコード1をもとに、
入力された整数を4で割った余りを表示する
プログラムを作成してください。
(制限時間20分)

サンプルコード2

3回「neko」と
表示するプログラム

実行結果

```
C:\WINDOWS\system32\cmd.exe
neko
neko
neko
続行するには何かキーを押してください
```

```
1  #include<stdio.h>
2
3  int main(void)
4  {
5      int n=0;
6
7      while (n < 3) {
8          printf("neko\n");
9          n++;
10     }
11     return 0;
12 }
```

while文

条件式を満たしている間ずっと
処理の内容をし続ける。

条件式のなかで「break」を使うと
ループをぬけだせる。

while(1)と書くと無限ループする。

```
while(条件式)
{
    処理
}
```

インクリメント・デクリメント

iを変数とすると、 $i++$, $++i$ をインクリメント
 $i--$, $--i$ をデクリメント と呼ぶ。

$i++$ は $i=i+1$ と同じ意味。つまり、iに1を足すということ。
 $i--$ も同様。

繰り返し文の時に使用されることが多い。

i++とi--の違い

先に加えるか、後に加えるかの違い。
代入時に大きく影響を及ぼす。

例) `x=i++;` と `x=++i;`

`int x,i=0;`

`x=i++;`

xにi(=0)を代入してからiに1を加える
x=0 , i=1

`x=0;`

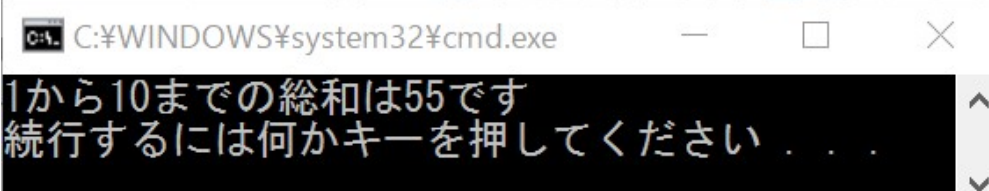
`x=++i;`

iに1を加えてからxにi(=1)を代入する
x=1 , i=1

サンプルコード3

1から10までの数の
総和を求める
プログラム

```
1  #include<stdio.h>
2
3  int main(void)
4  {
5      int n = 10, sum = 0;
6      int i;
7
8      for (i = 0; i <= n; i++) {
9          sum += i;
10     }
11     printf("1から10までの総和は%dです\n", sum);
12
13     return 0;
14 }
```



C:\WINDOWS\system32\cmd.exe

1から10までの総和は55です
続行するには何かキーを押してください . . .

for文

繰り返し文の一種。

()内に

初期化式（変数の初期値）

条件式（変数が条件を満たす間繰り返す）

変化式（処理後、変数の値を変化させる）

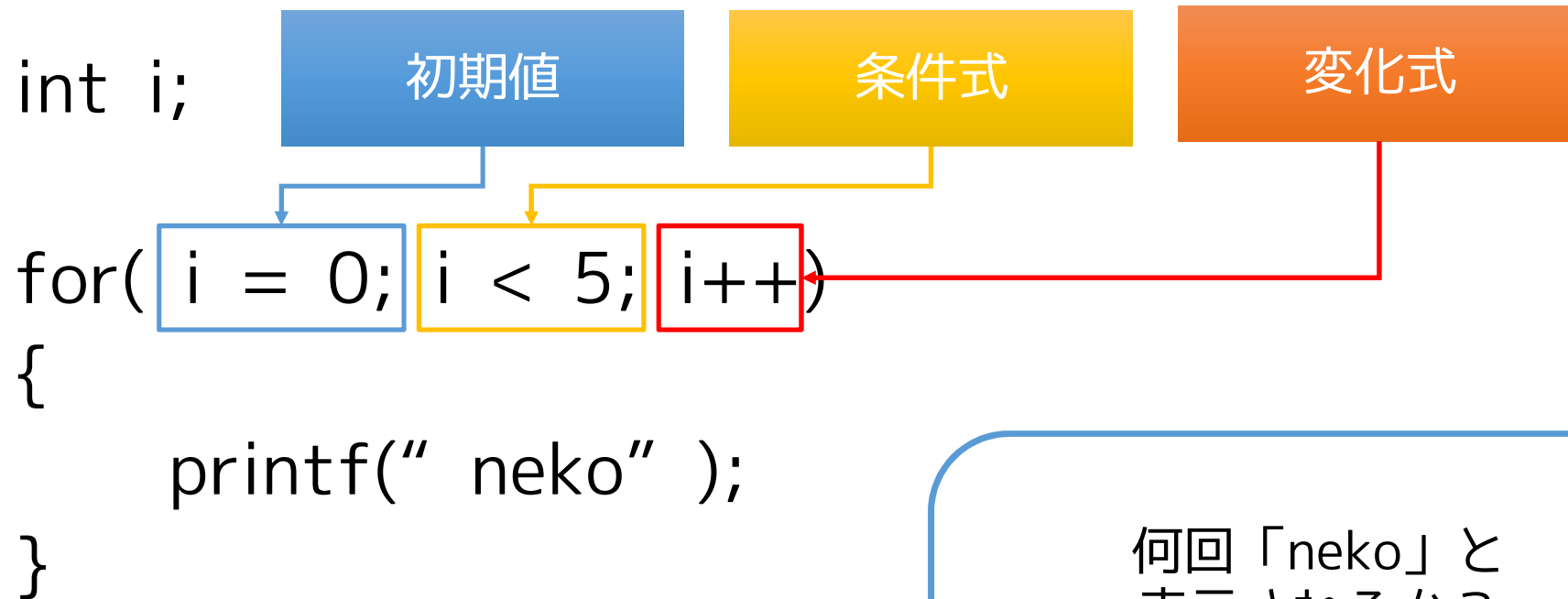
の3つの式を必要とする。

例) 変数 i が5以下の間は{ }内の処理を繰り返す

```
int i;  
  
for( i = 1; i <= 5; i++)  
    {  
        処理  
    }
```

初期化式 条件式 変化式

for文



何回「neko」と
表示されるか？

算術演算子と代入演算子

算術演算子と代入演算子を組み合わせることで、ソースコードを短縮することができる。

演算子	種類	例	意味
<code>+=</code>	加算代入	<code>a += b</code>	<code>a = a + b</code>
<code>-=</code>	減算代入	<code>a -= b</code>	<code>a = a - b</code>
<code>*=</code>	乗算代入	<code>a *= b</code>	<code>a = a * b</code>
<code>/=</code>	除算代入	<code>a /= b</code>	<code>a = a / b</code>

確認問題2

サンプルコード3を参考に、入力された数字nの階乗を求めるプログラムを作成してください。

(制限時間 30分)

ヒント

初期化の値は？

* = を使って...

サンプルプログラム4

入力された数字の
平方根と3乗を
求めるプログラム

`scanf(" %lf" ,&x);`

に注意

(double型を入力するには
%lfを使用する)

```
1  #include<stdio.h>
2  #include<math.h>
3
4  #define N 3
5
6  int main(void) {
7      double x,a,b;
8
9      printf("正の整数を入力してください\n");
10     scanf("%lf", &x);
11
12     a = sqrt(x);
13     b = pow(x, N);
14
15     printf("平方根 %f %d乗 %f\n", a, N, b);
16     return 0;
17
18 }
```

数学関数<math.h>

#include<math.h>を上部に書くと、
数学に関する関数を使用できる。

例)

sqrt(x); x の平方根をもとめる。

pow(x,y); x^y をもとめる。

他にも三角関数なども使用できる。

#define

コードの上部に、

#define 定数名 値

の順番でかくことで、定数を作ることができる。

円周率や、重力加速度などの定数などに用いることがある。

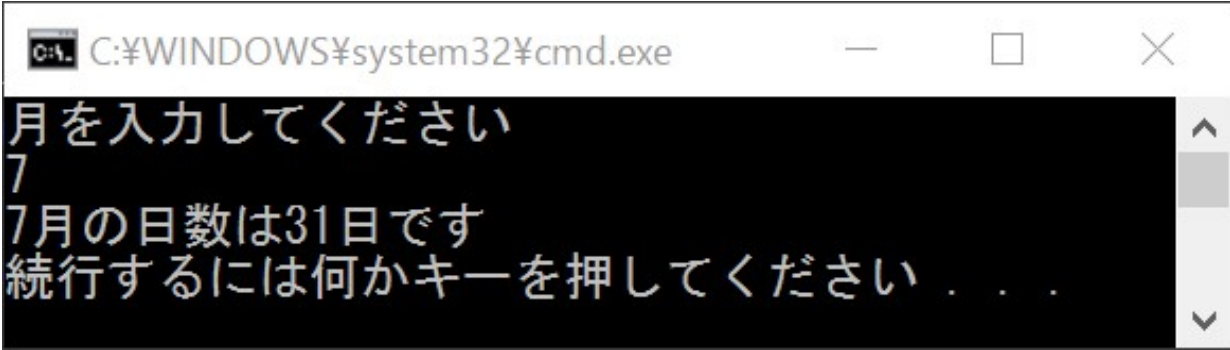
また、定数の値を変更しやすくするというメリットがある。

練習問題1

月の日数を表示するプログラム

(参考) 31日 ... 1月、3月、5月、7月、8月、10月、12月
30日 ... 4月、6月、9月、11月
28日 ... 2月

例) 7月

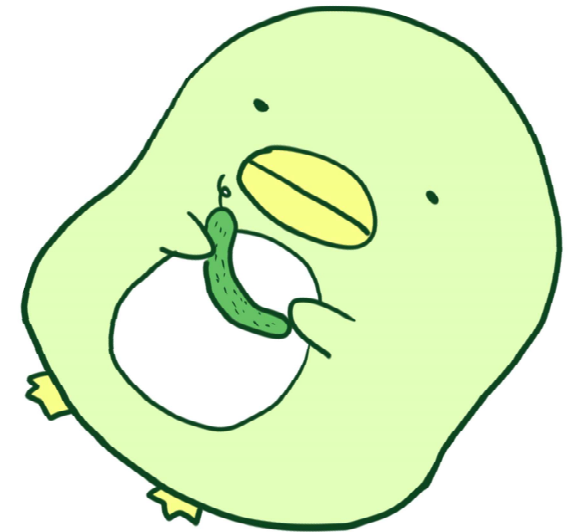


```
C:\WINDOWS\system32\cmd.exe
月を入力してください
7
7月の日数は31日です
続行するには何かキーを押してください . . .
```

練習問題2

お父さんペンギンのベン、お母さんペンギンのリリ、子供ペンギンのケイがいる。昨日は3羽で合計20匹のサカナを食べた。リリはケイより2匹多く食べ、ベンはリリより1匹多く食べた。ベン、リリ、ケイはそれぞれ何匹のサカナを食べたかを計算して表示するプログラムを作成せよ。

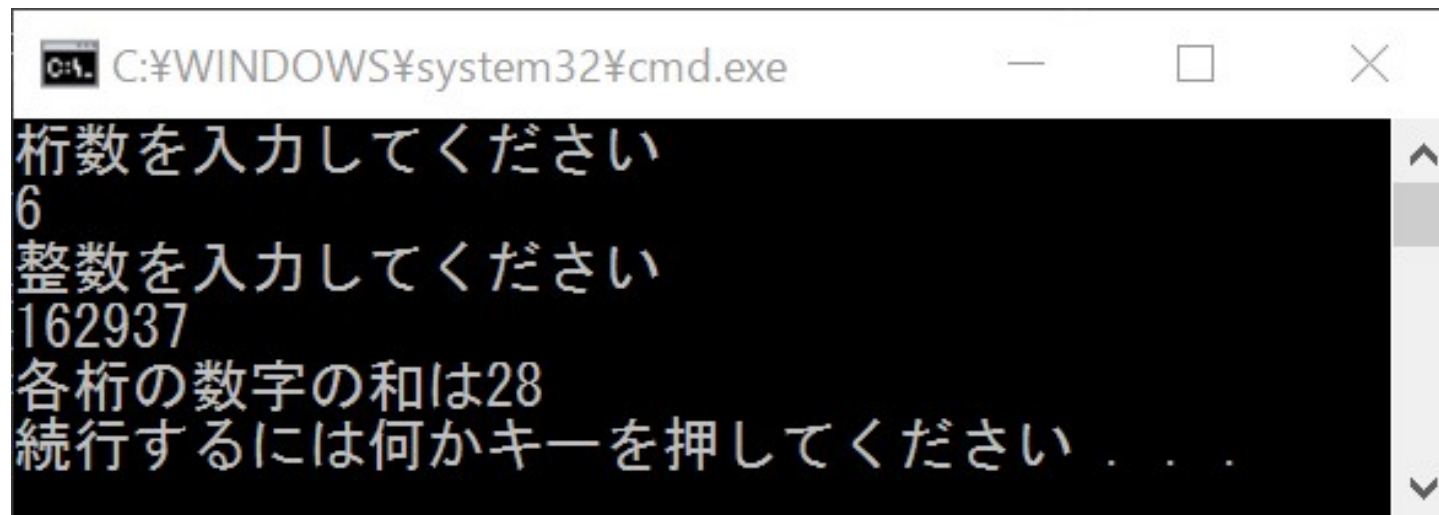
（連立方程式を手計算して答えだけ表示するのはNG）



練習問題3

n桁整数の各桁の数字の和を求めるプログラム

例)



```
C:\¥WINDOWS¥system32¥cmd.exe
桁数を入力してください
6
整数を入力してください
162937
各桁の数字の和は28
続行するには何かキーを押してください . . .
```


ゴリラ問題

次の式の第100000項まで求めよ

$$2\left(\frac{2}{1} - \frac{2}{3} + \frac{4}{3} - \frac{4}{5} + \frac{6}{5} - \frac{6}{7} + \frac{6}{7} - \frac{8}{9} + \dots\right)$$

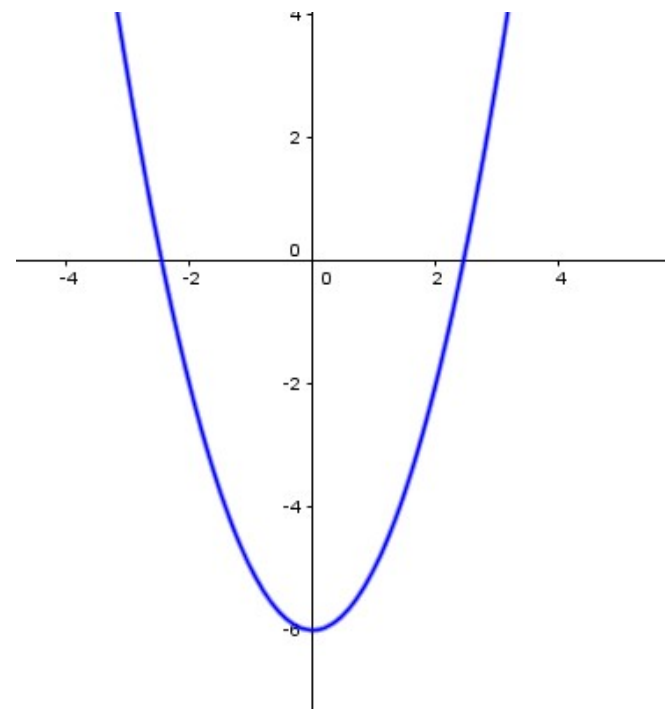
円周率に近似できたら正解. ウォリスの式という

スーパーゴリラ問題

- 数学関数 `double sqrt(double x)` を実装してみよう
終了条件は $|x - x_0 * x_0| < 0.001$ を使えばよい
 x_0 は暫定の解

絶対値は数学関数
`fabs(x)` を使えばよい

二次方程式の解を
求める問題に帰着できる
 $f(u) = u^2 - x = 0$



スーパーゴリラ問題ヒント

区間を狭めていくことで
解を求める方法がある

終了条件まで区間を半分にする
操作を繰り返せばよい

区間に解があるかどうかは
中間値の定理を使えば

$f(a) * f(b) < 0$ 解あり

$f(a) * f(b) > 0$ 解なし
ということが分かる。

