工業情報数理

１学期中間

担当者：松井（自称　昨日まで２０）

Practice makes perfect

「**継続は力なり**」に相当します。直訳すると「**練習は完璧を作る（完璧にする）**」ですが、そのままでもニュアンスは掴めると思います。

英語で「習うより慣れろ」

Experience is the best teacher.

→経験は最良の先生である

2 簡単なプログラム

図1は，画面に「ELIZABETH」と出力するプログラム[4]である。このプログラムからCのいくつかの特徴を学ぼう。

```
01  /* Cプログラムの例 */
02  #include<stdio.h>
03  int main(void)[5]
04  {
05      printf("ELIZABETH¥n"[6]);
06      return 0;
07  }
```

(a) プログラム

```
ELIZABETH
```

(b) 出力結果

▲図1　Cプログラムの例

A 文字　　C のプログラムは原則として小文字で記述する。ただし，行 05 の "ELIZABETH" は画面に出力するデータなので，大文字を用いてもよい。

```
printf("ELIZABETH¥n ");
```

"ELIZABETH¥n "

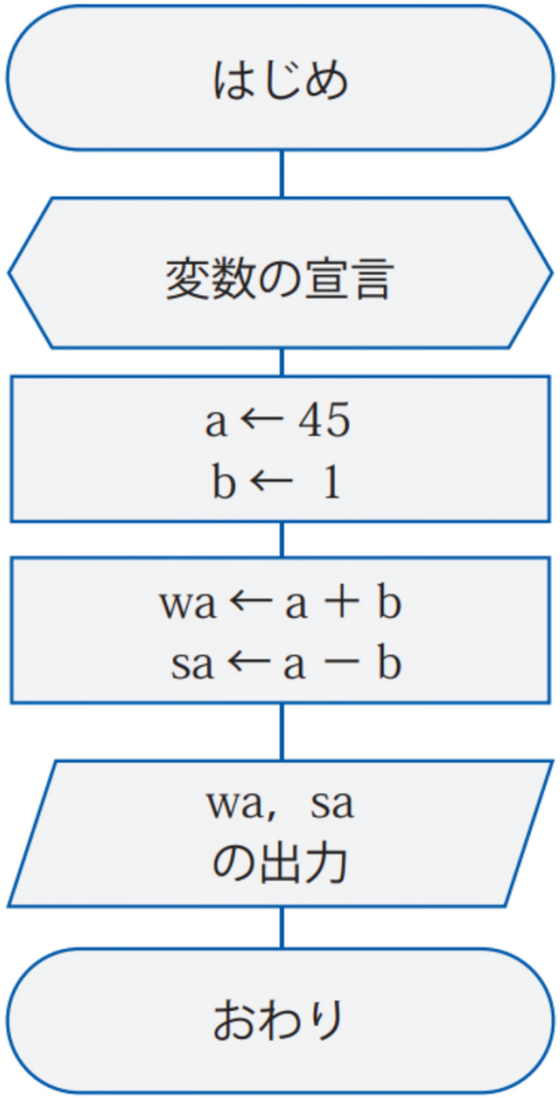
¥n　　改行

```c
#include <stdio.h>
int main(void)
{
    printf("あ");
    printf("い¥n¥n");
    printf("う");
    printf("¥n");
    printf("¥n");
    return 0;
}[EOF]
```

```
あい

う

続行するには何かキーを押してください
```

1 整数型データの取り扱い

```
01  /* 例題 1 */
02  #include<stdio.h>
03  int main(void)
04  {
05      int a, b, wa, sa;
06      a = 45;
07      b = 1;
08      wa = a + b;
09      sa = a - b;
10      printf("a + b = %d¥n", wa);
11      printf("a - b = %d¥n", sa);
12      return 0;
13  }
```



▲流れ図

```
型指定子　変数１，変数２，変数３，　……；
```

[例]　int a, b, wa, sa ;

表２に，おもな型指定子を示す。

▼表２　型指定子の例

型指定子		バイト長[1]	表現範囲
キャラ char	文字型	1	JIS コードで表現される１文字， または－128 ～ ＋127 までの整数
イント int	整数型	Tips 4	－2 147 483 648～＋2 147 483 647までの整数
ロング long	倍長整数型	4	－2 147 483 648～＋2 147 483 647までの整数
フロート float	単精度実数型	4	約-10^{+38}～約-10^{-38}， 約10^{-38}～約10^{+38}までの実数，および0
ダブル double	倍精度実数型	8	約-10^{+308}～約-10^{-308}， 約10^{-308}～約10^{+308}までの実数，および0

イント
int　　整数型

Tips
4

−2 147 483 648～＋2 147 483 647までの整数

２進数↵

１ビット→０と１の２通り↵

２ビット→００、０１、10、11　の４通り↵

↵

ビット↵	計算↵	何通り↵	正の数値のみ↵	負の数値のみ↵	両方になると↵
1↵	2^1↵	2↵	0～1↵	-2～-1↵	↵
2↵	2^2↵	4↵	0～3↵	-4～-1↵	-2～-1～0～1↵
3↵	2^3↵	8↵	0～7↵	-8～-1↵	-4～-1～0～3↵
4↵	2^4↵	16↵	0～15↵	-16～-1↵	-8～-1～0～7↵
5↵	2^5↵	32↵	0～31↵	-32～-1↵	-16～-1～0～15↵
6↵	2^6↵	64↵	0～63↵	-64～-1↵	-32～-1～0～31↵
7↵	2^7↵	128↵	0～127↵	-128～-1↵	-64～-1～0～63↵
8↵	2^8↵	256↵	0～255↵	-256～-1↵	-128～-1～0～127↵

1バイト（Byte）＝8ビット

2バイト（Byte）＝16ビット

3バイト（Byte）＝24ビット

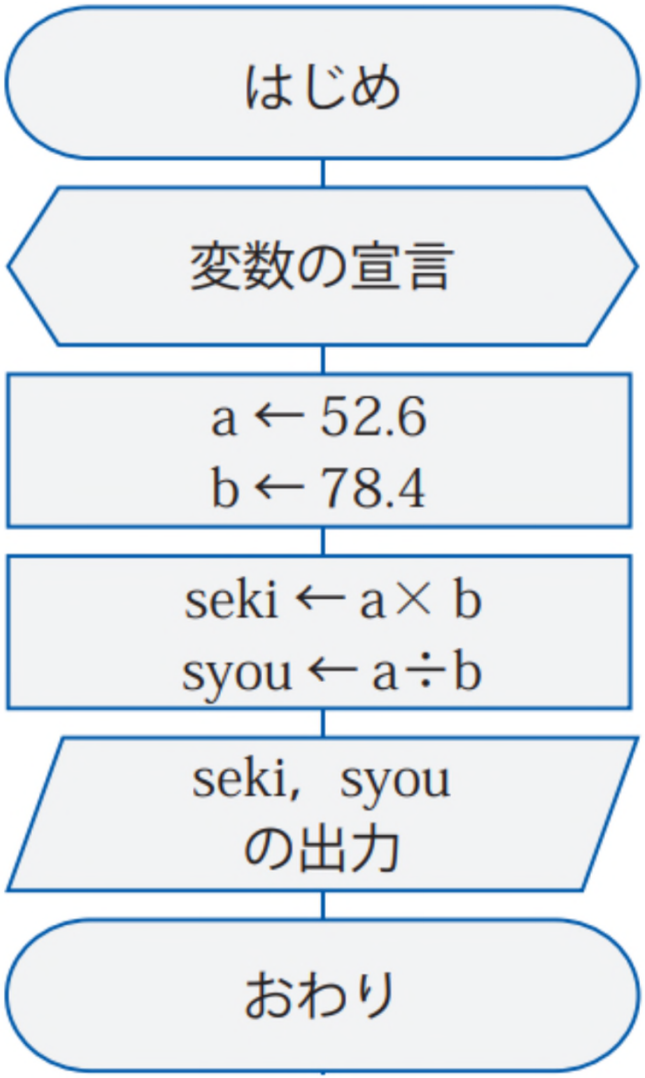
4バイト（Byte）＝32ビット

$2^{32}=$

フロート float	単精度実数型	
4	約-10^{+38}～約-10^{-38}， 約10^{-38}～約10^{+38}までの実数，および0	

2 実数型データの取り扱い

```
01  /* 例題 2 */
02  #include<stdio.h>
03  int main(void)
04  {
05      float a, b, seki, syou;
06      a = 52.6;
07      b = 78.4;
08      seki = a * b;
09      syou = a / b;
10      printf("a * b = %7.2f¥n", seki);
11      printf("a / b = %7.2f¥n", syou);
12      return 0;
```



▲流れ図

▼表 6　フィールド幅指定子の例

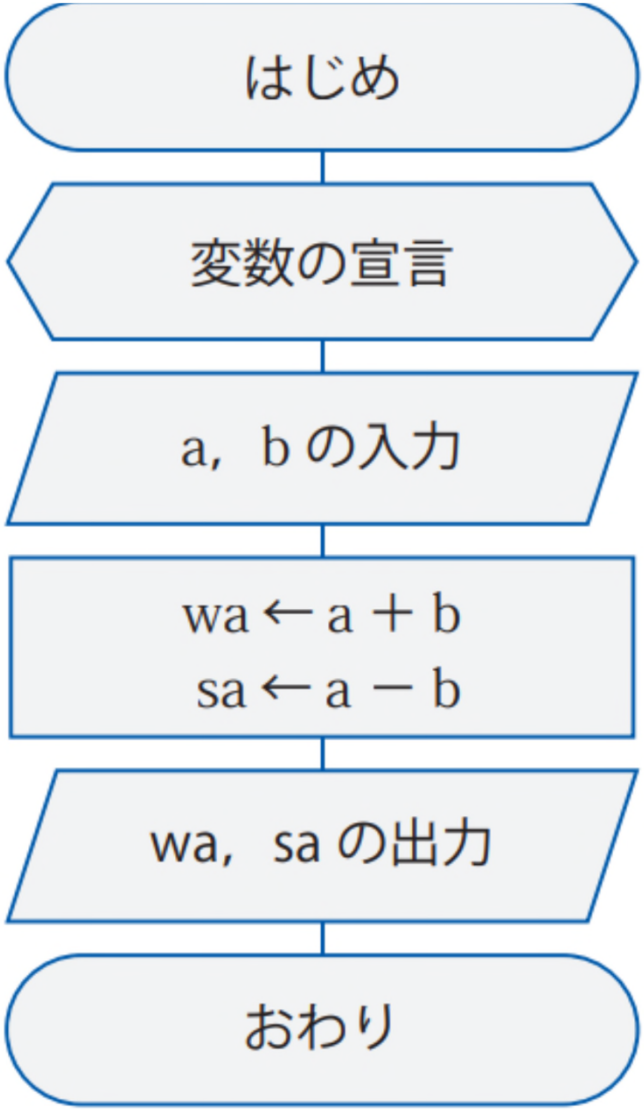
	変換指定子	出　力	意　味
整数型（データが1234の場合）	%d	1234	データの大きさに応じたけた数で出力する。
	%7d	␣␣␣1234	7 けたを確保し，右詰めで出力する。
	%2d	1234	指定子がデータより小さくても必要なけた数が確保される。
実数型（データが123.45の場合）	%f	123. 450000	データの大きさに応じたけた数で出力する。
	%13f	␣␣␣123. 450000	小数点を含んで 13 けたを確保し，右詰めで出力する。
	%11.3f	␣␣␣␣123. 450	小数点を含んで 11 けたを確保し，小数点以下 3 けたまでを右詰めで出力する。
	%7.1f	␣␣123. 5	小数点を含んで 7 けたを確保する。小数点以下の指定子がデータより小さいと小数第 2 位を四捨五入し，右詰めで出力する。
	%.1f	123. 5	整数部は必要幅で，小数点第 2 位を四捨五入して出力する。

整数型（データが1234の場合）	%d	1234	データの大きさに応じたけた数で出力する。
	%7d	␣␣␣1234	7けたを確保し，右詰めで出力する。
	%2d	1234	指定子がデータより小さくても必要なけた数が確保される。

実数型 （データが 123.45の場合）	%f	123. 450000	データの大きさに応じたけた数で出力する。
	%13f	␣␣␣123. 450000	小数点を含んで13けたを確保し，右詰めで出力する。
	%11.3f	␣␣␣␣123. 450	小数点を含んで11けたを確保し，小数点以下3けたまでを右詰めで出力する。

4 キーボードからのデータ入力

```
#include<stdio.h>
int main(void)
{
    int a, b, wa, sa;
    scanf("%d",&a);
    scanf("%d",&b);
    wa = a + b;
    sa = a - b;
    printf("a + b = %d¥n", wa);
    printf("a - b = %d¥n", sa);
    return 0;
}
```


はじめ
変数の宣言
a，b の入力
wa ← a ＋ b
sa ← a － b
wa，sa の出力
おわり

```
scanf（" 書式文字列"❶,  &変数 1,  &変数 2,  ……）;
```

[例]　scanf("%d",&a);

10進数として読み取ったデータを変数aに代入する

```c
#include <stdio.h>

int main(void)
{
    int a, b, wa, sa;

    scanf("%d",&a);
    scanf("%d",&b);

    wa = a + b;
    sa = a - b;
    printf("a + b = %d¥n", wa);
    printf("a - b = %d¥n", sa);
    return 0;
}[EOF]
```

1

2

$$a + b = 3$$

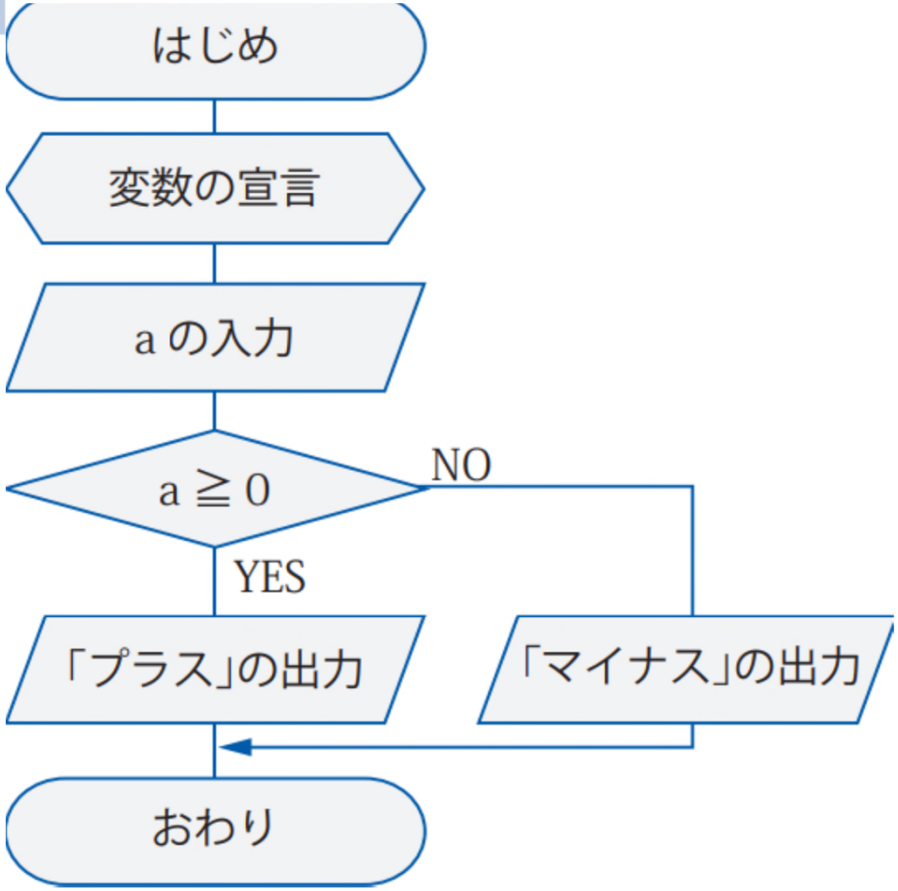
$$a - b = -1$$

```c
#include <stdio.h>

int main(void)
{
    int a, b, wa, sa;
printf("何か数値を入力してください→a=");
    scanf("%d",&a);

printf("何か数値を入力してください→b=");
    scanf("%d",&b);

    wa = a + b;
    sa = a - b;
    printf("a + b = %d¥n", wa);
    printf("a - b = %d¥n", sa);
    return 0;
```

```
何か数値を入力してください→a=1
何か数値を入力してください→b=2
a + b = 3
a - b = -1
```

1 if 文による分岐

```
/* 例題 5 */
#include<stdio.h>
int main(void)
{
    int a;
    scanf("%d",&a);
    if(a >= 0){
        printf(" プラス ¥n");
    }
    else{
        printf(" マイナス ¥n");
    }
    return 0;
}
```



```
if（条件）{
        条件が真の場合に実行する文１
        条件が真の場合に実行する文２
                ⋮
}
else {
        条件が偽の場合に実行する文１
        条件が偽の場合に実行する文２
                ⋮
}
```

```c
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d",&a);
    if(a >= 0) {
        printf("プラス ¥n");
    }
    else{
        printf("プラス ¥n");
    }
    return 0;
}[EOF]
```

3
プラス

−1

マイナス

演算子	記　号	使用例	意　味
関係演算子	＜	a＜b	a は b より小さい。
	＞	a＞b	a は b より大きい。
	＞＝	a＞＝b	a は b より大きいか等しい。
	＜＝	a＜＝b	a は b より小さいか等しい。
等価演算子	＝＝	a＝＝b	a と b は等しい。
	！＝	a！＝b	a と b は等しくない。

▼表９　論理演算子と単項演算子

	記号	使用例	意味
論理演算子	&&	x && y	xとyの論理積❶
	||	x || y	xとyの論理和❷
単項演算子	!	! x	xの否定

▼表 10　演算子の優先順位

順位	演算子
1	( )
2	!, ++, --❸
3	*, /, %
4	+, -
5	<=, <, >=, >
6	==, !=
7	&&
8	||
9	=, +=, -=, *=, /=, %=

```
   int a, b, s, d;
   s = 0;
   a = 5;
   b = 2;
   s = s + a;
printf("1kaime no s = %d¥n", s);
   s = s + 1;
   s = s + 2;
   s = s + 3;
printf("2kaime no s = %d¥n", s);
   d = s;
   d= d - b;
printf("1kaime no d = %d¥n", d);
   d= d - 1;
   d= d - 2;
printf("2kaime no d = %d¥n", d);
```

1kaime no s ANS.[5]

2kaime no s ANS.[11]

1kaime no d ANS.[9]

2kaime no d ANS.[6]

問題5　次のプログラムは、半径rを入力して、面積sと円周xを計算して求めるものである。プログラム中の①②③を考えなさい。円周率は3.14とする。　【知識･技術】

```
#include <stdio.h>

int main(void)
{
  float r,s,x;
  printf("半径ｒ＝");
  ①("%f",&r);
  s=3.14*②;
  x=2*3.14*r;
  ③ ("面積ｓ＝%f¥n",s);
  ③ ("円周ｘ＝%f¥n",x);
return 0;
}
```

出力結果

```
半径ｒ＝1
面積ｓ＝3.140000
円周ｘ＝6.280000
```

① ANS.[scanf]

② ANS.[r*r]

③ ANS.[printf]