

## 第3回 計算機基礎実習II

座席指定はありません。

Linux を起動して下さい。

計算機基礎実習II 2018 のウェブページから、以下の課題に自力で取り組んで下さい。

第2回の復習課題(rev02)

第3回の基本課題(base03)

## 第2回課題の回答例

### ex02-2.c

```
include <stdio.h>

int main {
  int l      /* 一辺の長さ */
  int v, s; /* 体積 v と表面積 s */

  scanf("%d", &l)
  printf(立方体の一辺の長さを入力して下さい: );

  printf("一辺の長さが %d の立方体の体積は l*l*l です。\\n" l, v);

  s = 6 * (a*2);
  print("一辺の長さが %d の立方体の表面積は %d です。\\n", l, s);

  return 0;
```

### cc ex02-2.c -o ex02-2

```
a00007@s01cd0542-160:~$ cc ex02-2.c -o ex02-2
ex02-2.c:1:9: error: expected '=', ',', ';', 'asm' or
'__attribute__' before '<' token
include <stdio.h>
      ^
ex02-2.c:8:4: error: stray '\347' in program
  printf(立方体の一辺の長さを入力して下さい: );
      ^
ex02-2.c:8:4: error: stray '\253' in program
ex02-2.c:8:4: error: stray '\213' in program
:
```

ex02-2.c:~ が1つのエラー

## ex02-2.c をデバッグ(debug)する

ソースファイルの先頭から順にデバッグする

```
a00007@s01cd0542-160:~$ cc ex02-2.c -o ex02-2
ex02-2.c:1:9: error: expected '=', ',', ';', 'asm' or
'__attribute__' before '<' token
include <stdio.h>
      ^
```

before '<': <の前で

expected ~ : こんなのが期待されてるんですけど...

ソースコードのこのへんで

ex02-2.c というファイルの1行目の9文字目で error 発生

## ex02-2.c

```
#include <stdio.h>

int main {
    int l    /* 一辺の長さ */
    int v, s; /* 体積 v と表面積 s */

    scanf("%d", &l)
    printf(立方体の一辺の長さを入力して下さい: );

    printf("一辺の長さが %d の立方体の体積は 1*1*1 です。\\n" l, v);

    s = 6 * (a*2);
    print("一辺の長さが %d の立方体の表面積は %d です。\\n", l, s);

    return 0;
}
```

## ex02-2.c

```
#include <stdio.h>

int main() {
    int l    /* 一辺の長さ */
    int v, s; /* 体積 v と表面積 s */

    scanf("%d", &l)
    printf(立方体の一辺の長さを入力して下さい: );

    printf("一辺の長さが %d の立方体の体積は 1*1*1 です。\\n" l, v);

    s = 6 * (a*2);
    print("一辺の長さが %d の立方体の表面積は %d です。\\n", l, s);

    return 0;
}
```

```
a00007@s01cd0542-160:~$ cc ex02-2.c -o ex02-2
ex02-2.c:3:10: error: expected '=', ',', ';', 'asm' or
'__attribute__' before '{' token
int main {
      ^
```

ソースコードのこのへんで

ex02-2.c というファイルの3行目の10文字目で error 発生

ex02-2.c の main() 関数の中の...

```
a00007@s01cd0542-160:~$ cc ex02-2.c -o ex02-2
```

```
ex02-2.c: In function 'main':
ex02-2.c:5:4: error: expected '=', ',', ';', 'asm' or
'__attribute__' before 'int'
  int v, s;      /* 体積 v と表面積 s */
  ^
```

```
ex02-2.c:7:17: error: 'l' undeclared (first use in this function)
  scanf("%d", &l)
                ^
```

```
ex02-2.c:7:17: note: each undeclared identifier is reported only once
for each function it appears in
```

5行目の int の前???

~ undeclared : そんなの知りません (宣言されてません) けど...

```
#include <stdio.h>
int main() {
  int l; /* 一辺の長さ */
  int v, s; /* 体積 v と表面積 s */
```

## ex02-2.c

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

```
int main() {
  int l; /* 一辺の長さ */
  int v, s; /* 体積 v と表面積 s */

  scanf("%d", &l);
  printf("立方体の一辺の長さを入力して下さい: ");
```

```
  printf("一辺の長さが %d の立方体の体積は l*l*l です。\\n", l, v);
```

```
  s = 6 * (a*2);
  print("一辺の長さが %d の立方体の表面積は %d です。\\n", l, s);
```

```
  return 0;
```

## ex02-2.c

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

```
int main() {
  int l; /* 一辺の長さ */
  int v, s; /* 体積 v と表面積 s */
```

```
  scanf("%d", &l);
  printf("立方体の一辺の長さを入力して下さい: ");
```

```
  printf("一辺の長さが %d の立方体の体積は l*l*l です。\\n", l, v);
```

```
  s = 6 * (a*2);
  print("一辺の長さが %d の立方体の表面積は %d です。\\n", l, s);
```

```
  return 0;
```

```
a00007@s01cd0542-160:~$ cc ex02-2.c -o ex02-2
```

```
ex02-2.c: In function 'main':
ex02-2.c:8:4: error: expected ';' before 'printf'
  printf("立方体の一辺の長さを入力して下さい: ");
  ^
```

```
ex02-2.c:8:4: error: stray '\347' in program
ex02-2.c:8:4: error: stray '\253' in program
ex02-2.c:8:4: error: stray '\213' in program
ex02-2.c:8:4: error: stray '\346' in program
ex02-2.c:8:4: error: stray '\226' in program
ex02-2.c:8:4: error: stray '\271' in program
  ;
```

8行目の printf の前???

'\343' や '\200' のような英数表記できないコードの  
バグの多くは、全角日本語文字 (列) に関するエラー  
(全角の空白文字も同様のエラーの原因となる)

```
a00007@s01cd0542-160:~$ cc ex02-2.c -o ex02-2
ex02-2.c: In function 'main':
ex02-2.c:11:80: error: expected '}', before 'l'
printf("一辺の長さが %d の立方体の体積は l*l*l です。 \n" l, v);
```

```
^
ex02-2.c:13:13: error: 'a' undeclared (first use in this function)
s = 6 * (a*2);
          ^
ex02-2.c:13:13: note: each undeclared identifier is reported only once
for each function it appears in
ex02-2.c:16:3: error: expected declaration or statement at end of input
return 0;
^
```

## ex02-3.c

```
#include <stdio.h>

int main() {
    int a;
    printf("整数を入力して下さい: ");
    scanf("%d", &a);

    printf("入力された値は %d です。 \n", a);
    return 0;
}
```

### ex02-1.c から修正する

```
#include <stdio.h>

int main() {
    int a;
    printf("整数 a を入力して下さい: ");
    scanf("%d", &a);

    printf("a^2 は %d です。 \n", a*a);
    printf("a^3 は %d です。 \n", a*a*a);
    return 0;
}
```

## ex02-5.c

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

int main() {
    int a;
    printf("a の値を入力して下さい ");
    scanf("%d", &a);

    printf("1^(-2) = %f\n", 1./(a*a));
    printf("1^(-1/2) = %f\n", 1./sqrt(a));
    printf("1^(1/2) = %f\n", sqrt(a));
    printf("1^(2) = %f\n", (double)a*a);

    return 0;
}
```

## いくつかのお知らせ (再)

### Visual Studio Code の使い方まとめ

(GUIからVisua Studio Code を起動する)

オンライン・プログラミング環境 (paiza.io)

「理工出席」アプリのテスト運用

## 第3回の内容

プログラミング言語Cの関数

main()関数

return 文

変数の値は有限

## 関数

数学の関数

プログラミング言語の関数 func(a) e.g. printf()

int a      func()      int v

## プログラミング言語の関数

プログラミング言語の関数      int a      func()      int v

入力がない場合もある      func()      int v

出力がない場合もある      int a      func()

別の入力や出力をする場合もある      int a      func()      int v

scanf()

printf()

## main() 関数

C言語で最初に呼び出される関数

実行可能なCプログラムを書く = main() 関数を定義する

return 文によって関数の値 (戻り値) を定義する

main という(何もしない)名前の関数

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

main は int 型の0を (戻り値として) 返す

main は int 型を (戻り値として) 返す

## 関数と返り値とreturn文

C言語の関数は「関数名() [...]」で定義される (いまはmain() だけ詳細は後日)

C言語の関数は「関数名()」で呼び出される(利用される)

C言語の関数は値をもつ (もたないこともある →)      int a      func()

return 文によって関数の値 (返り値) が決まる

scanf() 関数と printf() 関数も int 型の返り値をもつ

```
int a, n, m;

n = scanf("%d", &a);
m = printf("a is %d\n", a);
// n? m?
```

## 第3回の基本課題

### ex03-1.c

```
#include <stdio.h>
int main() {
    int a;
    scanf("%d", &a);
    return 2*a;
}
```

直前に実行したコマンドの返り値が \$? に入る

```
t180900@s01cd0542-160:~$ echo $? 変数 $? の値を表示する
2
```

### ex03-2.c

while 文の復習：繰り返し処理

この条件が成立しているあいだ繰り返す

{...} で囲まれたブロック  
を繰り返す

```
while (i > 0) {
    i *= 2;
    printf("%d\n", i);
}
```

## ex03-2.c : 変数値の範囲は有限

```
t180900@s01cd0542-160:~/kiso2-2018/ex03$ ./ex03-2
2
4
8
16
32
64
128
:
134217728
268435456
536870912
1073741824
-2147483648      2倍しているだけなのに突然負の値に！！
```

負の値になることも、それがこの数値になることも「情報処理の基礎」で学ぶ「2の補数表現」を理解すると、きちんと説明できます。