

2015年度 プログラミング講習

電気通信大学 MMA

やること

1. プログラムとは
2. C言語
3. 問題

プログラム

ソフトウェア

- ls
- itunes.exe
- firefox.exe

- GNU ls
- iTunes
- Firefox
- windows10
- Debian
- bash

機械語

```
[root@ins ~]# xxd -g1 -b /bin/ls | head -n 50
0000000: 01111111 01000101 01001100 01000110 00000010 00000001 .ELF..
0000006: 00000001 00000000 00000000 00000000 00000000 00000000 .....
000000c: 00000000 00000000 00000000 00000000 00000010 00000000 .....
0000012: 00111110 00000000 00000001 00000000 00000000 00000000 >.....
0000018: 10010000 01001000 01000000 00000000 00000000 00000000 .He...
000001e: 00000000 00000000 01000000 00000000 00000000 00000000 ..e...
0000024: 00000000 00000000 00000000 00000000 00000000 10100111 .....
000002a: 00000001 00000000 00000000 00000000 00000000 00000000 .....
0000030: 00000000 00000000 00000000 00000000 01000000 00000000 ....@.
0000036: 00111000 00000000 00001001 00000000 01000000 00000000 8...@.
000003c: 00011100 00000000 00011011 00000000 00000110 00000000 .....
0000042: 00000000 00000000 00000101 00000000 00000000 00000000 .....
0000048: 01000000 00000000 00000000 00000000 00000000 00000000 @.....
000004e: 00000000 00000000 01000000 00000000 01000000 00000000 ..@.@.
0000054: 00000000 00000000 00000000 00000000 01000000 00000000 ....@.
000005a: 01000000 00000000 00000000 00000000 00000000 00000000 @.....
0000060: 11111000 00000001 00000000 00000000 00000000 00000000 .....
0000066: 00000000 00000000 11111000 00000001 00000000 00000000 .....
000006c: 00000000 00000000 00000000 00000000 00001000 00000000 .....
0000072: 00000000 00000000 00000000 00000000 00000000 00000000 .....
0000078: 00000011 00000000 00000000 00000000 00000100 00000000 .....
000007e: 00000000 00000000 00111000 00000010 00000000 00000000 ..8...
0000084: 00000000 00000000 00000000 00000000 00111000 00000010 ....8.
000008a: 01000000 00000000 00000000 00000000 00000000 00000000 @.....
0000090: 00111000 00000010 01000000 00000000 00000000 00000000 8.@...
0000096: 00000000 00000000 00011100 00000000 00000000 00000000 .....
000009c: 00000000 00000000 00000000 00000000 00011100 00000000 .....
00000a2: 00000000 00000000 00000000 00000000 00000000 00000000 .....
00000a8: 00000001 00000000 00000000 00000000 00000000 00000000 .....
```

アセンブリ言語

```
[octal]trunc,NOR]% objdump -d /bin/ls | head -n 50
```

```
~/programming講習/mas
```

```
/bin/ls:      ファイル形式 elf64-x86-64
```

```
セクション .init の逆アセンブル:
```

```
0000000000402168 <_init>:
```

```
402168:      48 83 ec 08      sub    $0x8,%rsp
40216c:      48 8b 05 85 7e 21 00  mov    0x217e85(%rip),%rax      # 619ff8 <_fini+0x2080fc>
402173:      48 85 c0          test   %rax,%rax
402176:      74 05            je     40217d <_init+0x15>
402178:      e8 33 04 00 00   callq 4025b0 <__gmon_start__@plt>
40217d:      48 83 c4 08      add    $0x8,%rsp
402181:      c3              retq
```

```
セクション .plt の逆アセンブル:
```

```
0000000000402190 <_stunp_tounp_1@plt-0x10>
```

高級言語

- FORTRAN
- COBOL
- LISP
- BASIC
- C言語
- Python
- Haskell
- Ruby
- Java
- JavaScript ...

参考: <http://ja.wikipedia.org/wiki/プログラミング言語年表>

コンパイラ

例えば GNU Compiler Collection

```
$ gcc hello.c
```

```
$ ./a.out
```

```
hello
```

```
$
```


インタプリタ

```
$ ruby -e 'puts("hello")'  
hello  
$
```


プログラマ

- アプリケーション開発者
- システムソフトウェア開発者
- 組み込みファームウェア開発者
- ソフトウェア技術者
- 計算機科学者
- ソフトウェアアナリスト
- システムエンジニア ...

参考: <http://ja.wikipedia.org/wiki/プログラマ>

C言語

おやくそく

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

```
int main(void){ /*コメント*/  
    printf("Hello, World!\n");  
    return 0;    //コメント  
}
```

実行

```
$ gcc -o hello hello.c -std=c99
```

```
$ ./hello
```

```
Hello, World!
```

```
$
```

数学での関数

$$g(x) = \log(x) - \sin(x)$$

$$f(x) = g(x) + \cos(x)$$

Hello, World!

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

```
int main(void){  
    printf("Hello, World!");  
    printf("\n");  
    return 0;  
}
```

関数

$\text{sum}(x, y) = x + y$ (x, y : 整数)

$\text{sum}(2, 3) = 5$

```
int sum(int x, int y){  
    return x + y;  
}
```


数值出力

```
printf(“%d年度入学\n”, 2015);
```

変数

```
int main(void){  
    int num = 3;  
    int hoge = num;  
    printf(“%d\n”, hoge);  
    return 0;  
}
```

配列

```
int array[10];  
array[0] = 3;  
array[1] = 5;  
printf(“%d\n”, array[1]);
```

問1 swap

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

```
int main(void){
```

```
    int n = 2, m = 10;
```

```
    printf("n= %d, m= %d\n", n, m);
```

```
    /*ここにコードを書く*/
```

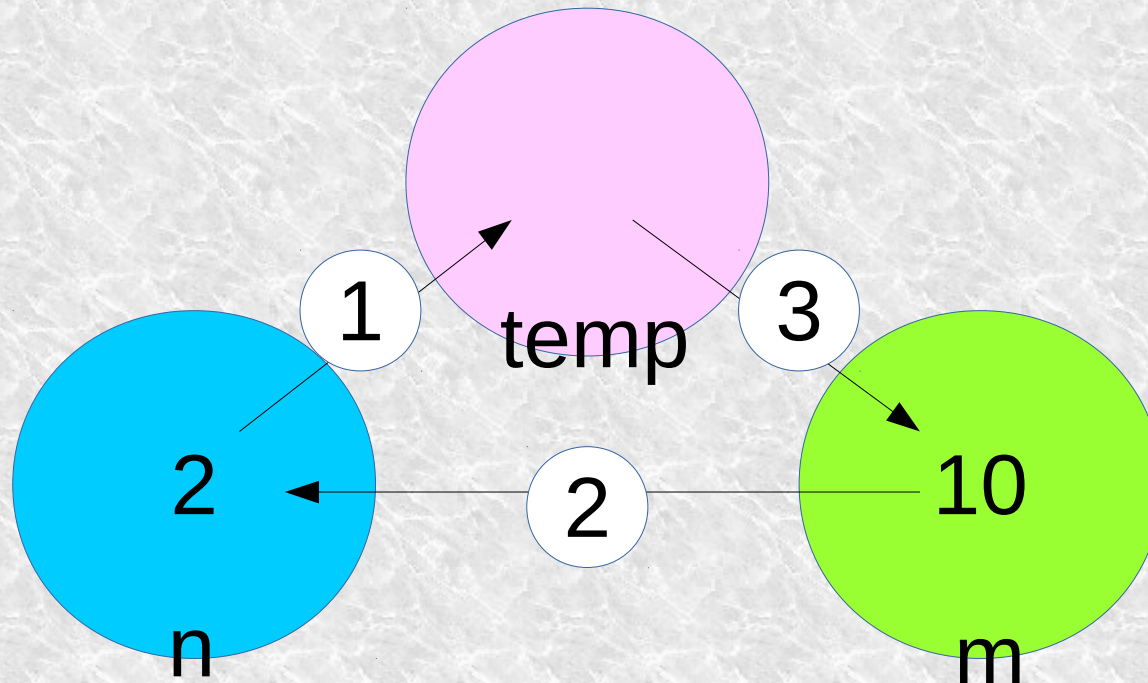
```
    printf("n= %d, m= %d\n", n, m);
```

```
    return 0;
```

```
}
```

問1 答

- ① `int temp = n;`
- ② `n = m;`
- ③ `m = temp;`



数值输入

```
int num;  
scanf("%d", &num);  
int array[3];  
scanf("%d", &array[0]);  
scanf("%d", &array[1]);  
scanf("%d", &array[2]);  
  
printf("%d\n", array[2]);
```

条件分岐

```
if( num < 10 ){ // (1)もしnumが10以下なら
    num = num * 100;

}else if( num < 100 ){
    // (2) (1)でないかつもしnumが100以下なら
    num = num * 10;

}else{ // (1), (2)でないとき
    num = num * 5;
}
```


問2 絶対値

```
#include<stdio.h>
//ここにabsの定義を書く
int main(void){
    int n;
    scanf("%d", &n);
    printf("%d\n", abs(n));
    return 0;
}
```

問2 答

```
int abs(int x){  
    if( x < 0 ){  
        x = x * (-1);  
    }  
    return x;  
}
```

反復

```
#include<stdio.h>
int main(void){
    for(int i = 0; i < 10; i = i + 1){
        printf(“%d\n”, i);
    }
    return 0;
}
```

問3 べき乗

整数 x , y を引数にとって
 x の y 乗を返す関数を定義

(C言語で \wedge という演算子は
違う意味で用いられている)

問3 答

```
int power(int x, int y){  
    int res = 1;  
    for(int i = 0; i < y; i = i + 1){  
        res = res * x;  
    }  
    return res;  
}
```

GNU Make

```
C = gcc
CFLAGS = -std=c99 -Wall
SOURCE = $(wildcard *.c)
all: $(basename $(SOURCE))
```

```
%.c:
    $(C) -o $@ $^ $(CFLAGS)
```

\$ make -k

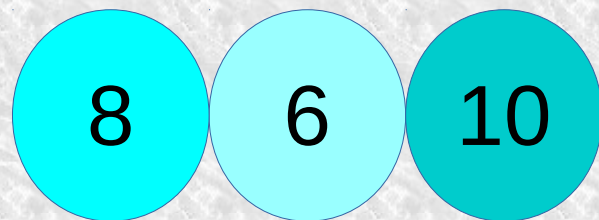
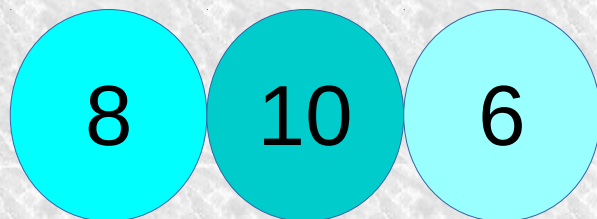
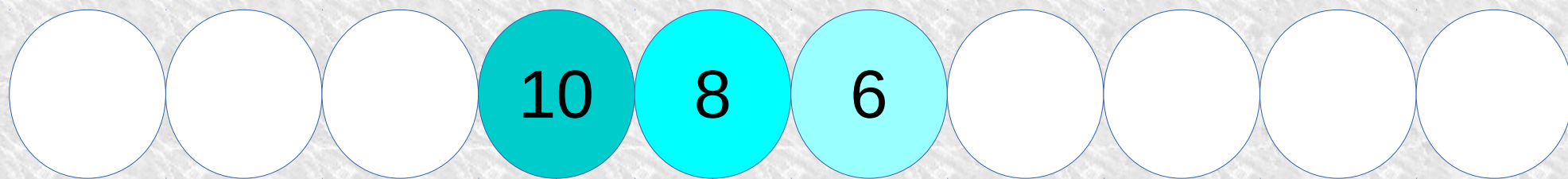
gcc -o hoge hoge.c -std=c99 -Wall

Makefile

問4 昇順ソート

```
#include<stdio.h>
void sort(int array[], int size){
}
int main(void){
    int array[10];
    for(int i = 0; i < 10; i = i + 1){
        scanf("%d", &array[i]);
    }
    sort(array, 10);
    for(int i = 0; i < 10; i = i + 1){
        printf("%d\n", array[i]);
    }
    return 0;
}
```


問4 ヒント



問4 答

```
void sort(int array[], int size){
    for(int i = 0; i < size; i = i+1){
        for(int k = 1; k < size - i; k = k+1){
            if( array[k] < array[k-1] ){
                int tmp = array[k];
                array[k] = array[k-1];
                array[k-1] = tmp;
            }
        }
    }
}
```