

# 計算の科学

論理演算とブール代数

2進数、8進数、16進数

データ操作とシフト、そして暗号

プログラミング

# 目的

- PC(パソコン)やスマートフォンなど、身近な情報機器を使って、「計算の科学」の初歩的かつ科学的な理解を促す

# 対象

- 中学生高学年から、高校生、一般計算の科学に興味のある人ならだれでも
- コンピュータの基礎知識や、アプリケーションの使い方などは、本セミナーの対象としない

※ コンピュータの基礎知識(Windows, ワード、エクセル、プレゼンテーションの使い方)については、実年世代を対象に別途開催を用意

# AGENDA

- 「計算の科学」セミナーの教材です
  - 1章： ブール代数
  - 2章： 8進数、16進数
  - 3章： 2進演算実習
  - 4章： プログラミング

# 必要な環境

- 人数に応じて準備が必要
- 座学(第1回) 100分  
power point, ホワイトボード、プロジェクタ
- プログラミング(第2回) 100分  
MAC/ iOS あるいはWINDOWS

# 第1章

## ● ブール代数

2進数

論理演算とブール代数



# ブール代数

- 2進数は、0と1だけからなる代数系  
ブール演算子はコンピューター 内部の演算そのもの

OR / AND / NOT 回路

# ブール代数の演算

OR

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

AND

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

NOT

0	1
1	0



# 掛け算は足し算の繰り返し

- 掛け算は足し算、割り算は引き算
  - 引き算の負数は、補数を足す
  - 補数はビットの反転
  - $2^n - 1$ による補数を1の補数という
  - 一般に、負数には2の補数を使う
  - 最左ビットは符号ビット

掛け算の概念には、倍(Times)、乗(Multiply)、積(Product)  
ここでの掛け算は、倍数のこと

# 2進数の引き算は補数の足し算

- ・ 1の補数の求め方: 各桁の1と0を反転する
- ・ 2の補数の求め方: 1の補数 + 1

0010 0100 (元の数 36)  
1101 1011 (1の補数 219)  
1101 1100 (2の補数 220)

```
10000 0000 (256)
-) 0010 0100 (36)
-----
1101 1100 (220)
```

0010 0100 + 1101 1100 => すべての桁が、0 --- 0

最上位桁上がり 1 は無視

# 第2章

- 8進数、16進数



# 2進数、8進数、16進数

10進数	2進数	8進数	16進数
0	0000 0000	0	0
1	0000 0001	1	1
2	0000 0010	2	2
3	0000 0011	3	3
4	0000 0100	4	4
5	0000 0101	5	5
6	0000 0110	6	6
7	0000 0111	7	7
8	0000 1000	10	8

10進数	2進数	8進数	16進数
9	0000 1001	11	9
10	0000 1010	12	A
11	0000 1011	13	B
12	0000 1100	14	C
13	0000 1101	15	D
14	0000 1110	16	E
15	0000 1111	17	F
16	0001 0000	20	10
17	0001 0001	21	11

# 2進数でオール1はメルセンヌ数

n	$2^n - 1$	2進数	8進数	16進数
2	3	0000 0000 0011	3	3
3	7	0000 0000 0111	7	7
4	15	0000 0000 1111	17	F
5	31	0000 0001 1111	37	1F
6	63	0000 0011 1111	77	3F
7	127	0000 0111 1111	177	7F
8	255	0000 1111 1111	377	FF
9	511	0001 1111 1111	777	1FF
10	1023	0011 1111 1111	1777	3FF
11	2047	0111 1111 1111	3777	7FF
12	4095	1111 1111 1111	7777	FFF
13	8191	1 1111 1111 1111	17777	1FFF

# 演習

- 次の10進数、2進数、8進数、16進数を変換せよ

10進数	2進数	8進数	16進数
	0110 0100		64
144		220	
	1010 1110		AE
192		300	
	1 0000 0000		100

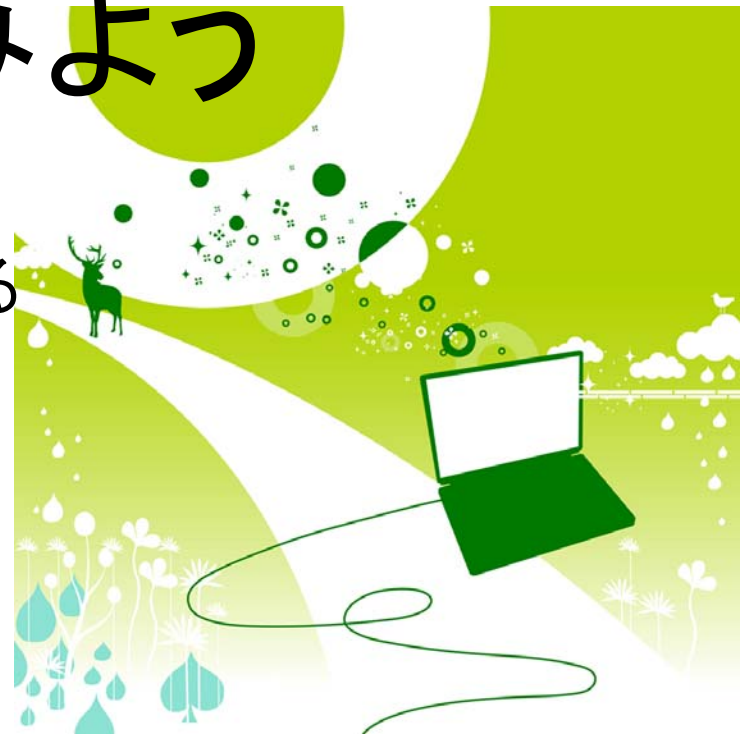
## 10進数、2進数、8進数、16進数の変換結果

10進数	2進数	8進数	16進数
100	0110 0100	144	64
144	1001 0000	220	90
174	1010 1110	256	AE
192	1100 0000	300	C0
256	1 0000 0000	400	100

# 第3章

## ● 演算をしてみよう

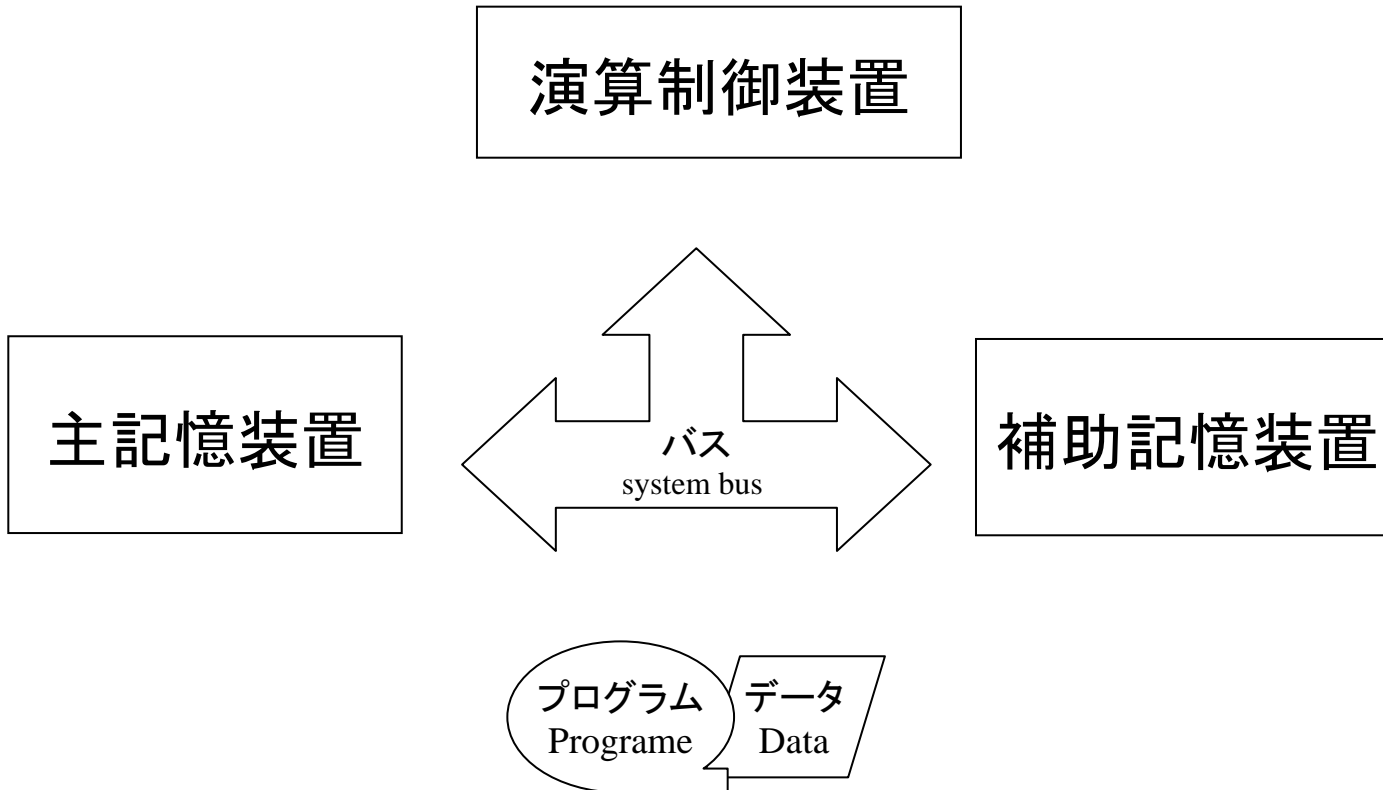
オセロ型計算機で演算する





# プログラム内蔵型コンピュータ

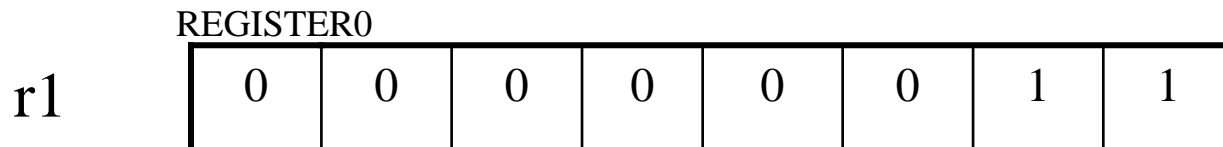
Stored program-computer



## 2進数による演算装置

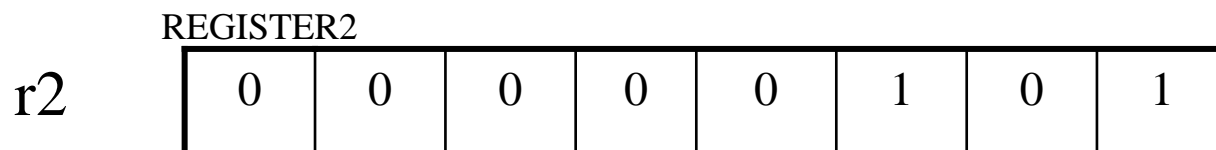
3 + 5 = 8 を計算する

m1 ← 3, m2 ← 5

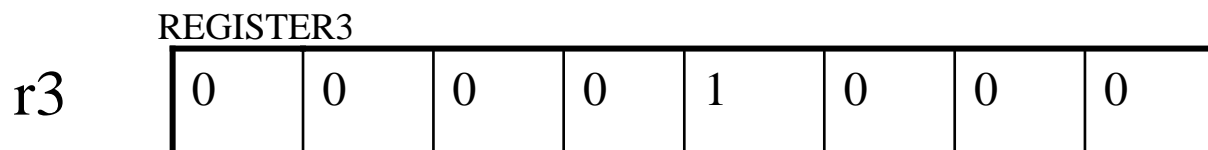


(機械語のシーケンス)

LDA m1, r1

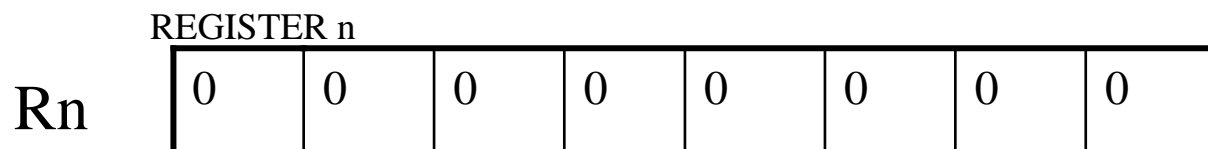


LDA m2, r2



ADD m1, m2, r3

STR r3, m3



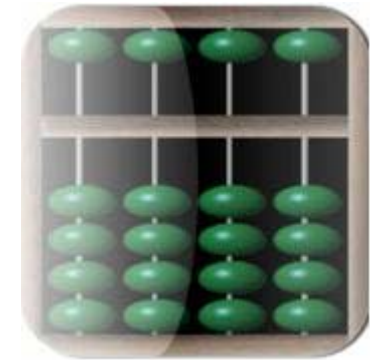
m3 ← 5 答え

# 桁シフトは2のべき乗

$$2 \times 2 = 2 + 2$$

$$2 \times 3 = 2 + 2 + 2$$

$$2 \times 4 = 2 + 2 + 2 + 2$$



10進数  $2 \times 2 = 2 + 2$

2進数  $0010 \times 0010 = 0010 + 0010$

$$0010$$

$$\underline{+ 0010}$$

$$00100$$

← 2の冪数は左へシフト

# 演習

- オセロ型計算機シミュレータを使い、次の計算をこなさい。

①  $3 \times 9$

②  $3 \times 78$

ヒント

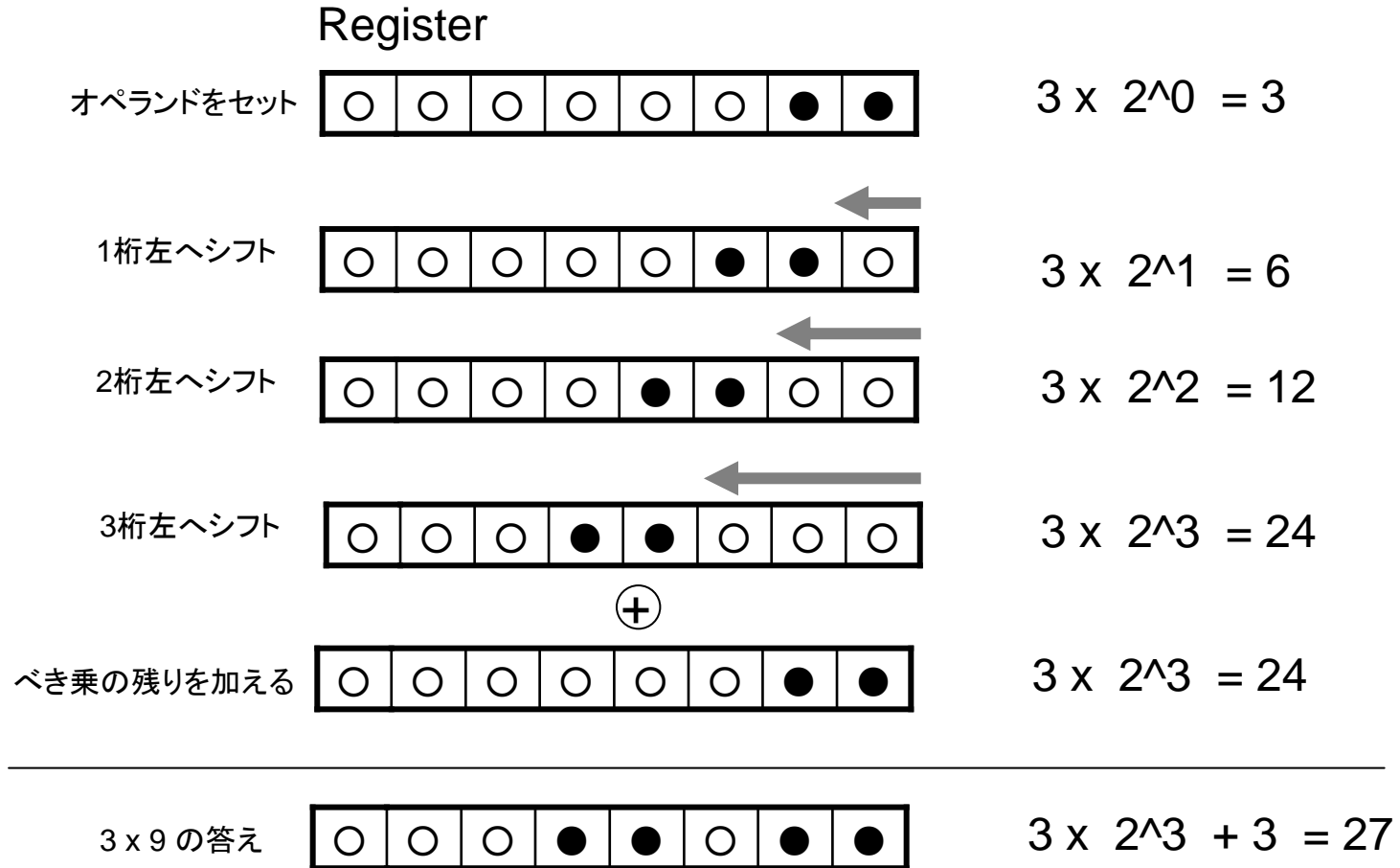
$$3 \times 9 = 3 \times (8 + 1)$$

$$3 \times 78 = 3 \times (2^6 + 2^3 + 2^2 + 2^1)$$

$$3 \times 9 = 3 \times (8 + 1)$$

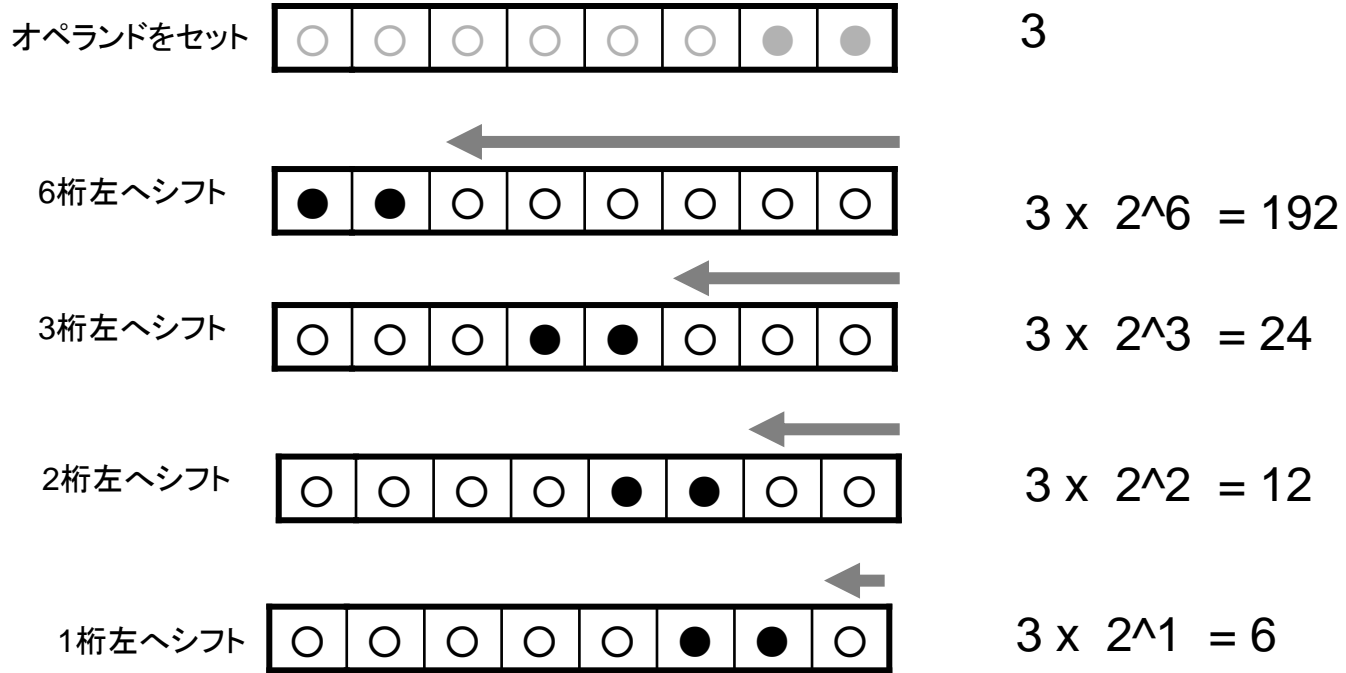
$$= (3 \times 2^3) + (3 \times 1) \rightarrow 0011000 + 0011 = 0011011$$

0011(3)を9回足さずに3回左にシフトした後、3を足すだけで結果が得られました。

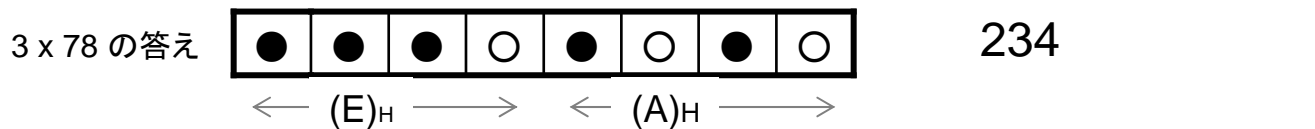


$$3 \times 78 = 3 \times (2^6 + 2^3 + 2^2 + 2^1)$$

Register



⊕



# 演習

2進数の理解のために、次の掛け算を、シフトで  
できることを証明せよ

$$49 \times 85 = 4165$$

2で割る

49  
24  
16  
6  
3  
1

2を掛ける

85  
~~170~~  
~~340~~  
~~680~~  
1360  
2720  
4165

## 答え

$$\begin{aligned} & 49 \times 85 \\ &= (1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0) \times 85 \\ &= (32 + 16 + 0 + 0 + 0 + 1) \times 85 \\ &= 2720 + 1360 + 0 + 0 + 0 + 85 \\ &= 4165 \end{aligned}$$

49を2進数であらわすと、 $2^3$ ,  $2^2$ ,  $2^1$  は表れから、  
85に $2^3$ を掛けた680、 $2^2$ を掛けた34、 $2^1$ を掛けた170を  
右の行で加える必要がない



# 第4章

## ● プログラミング

Object-c, Java, Pythonなどスクリプト言語に挑戦



# プログラミング演習

指定された言語を使用して、次の級数を求め、結果をディスプレイに印字せよ

演習1  $1 + 2 + 3 + \dots + 10$

演習2 2進数を10進数へ変換

演習3 10進数を2進数へ変換

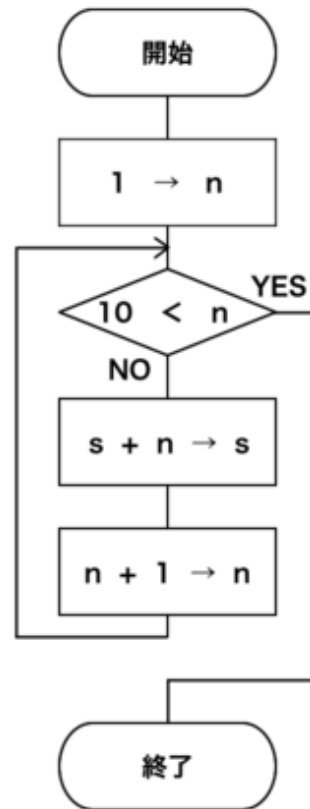
演習4  $1/1^2 + 1/2^2 + 1/3^2 \dots$

演習5  $1 + 1 + 1/2! + 1/3! + \dots$

# 演習1

- ①  $1 + 2 + 3 + \dots + 10$ を順に入力する。
- ② 入力されたデータをチェックし、エラーなら再入力を促す  
データ範囲チェックの条件：  
nは整数  
 $n_0 < n_1 < n_2 \dots < n_{10} \quad 0 \leq n_x \leq 10$   
ESCが入力されたら、無条件で終了
- ② 入力されたデータを加算して、一時保存する
- ③ 入力カウントが10になったら、合計値を表示し終了する

# フローチャート



# コーディング

- テキストエディタ上で、プログラムをかいてみる

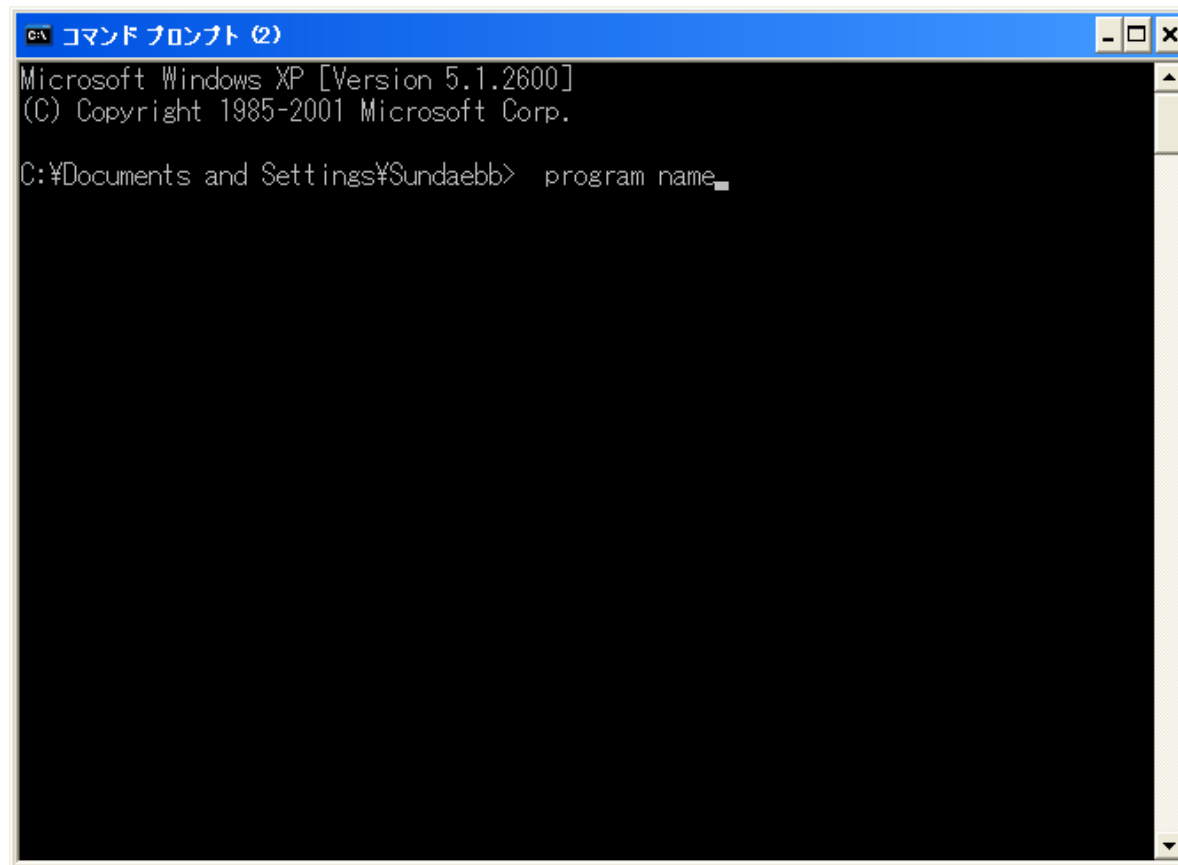
演習2 2進数を10進数へ変換

使用言語を C言語とする

# コーディング

```
/**  
/* 2進数→10進数変換 */  
/**  
#include <stdio.h>  
void main( )  
{  
    char c[128];  
    int i,n;  
  
    printf("¥n***2進数を10進数に変換します***");  
    printf("¥n2進数: "); fflush(stdout); gets(c);  
    i=0; n=0;  
    while ( 1 ) // 無限に繰り返す  
        {  
            if ( c[i]=='1' ) n=n+1; // 文字が1のとき1を加算  
            i=i+1; // 次の文字へ  
            if ( c[i]=='¥0' ) break; // 入力文字列の最後  
            n=n*2; // 次の桁へ  
        }  
  
    printf("¥n10進数=%d",n); }
```

# プログラムの実行



The image shows a screenshot of a Windows XP Command Prompt window. The title bar reads "コマンド プロンプト (C)". The window content displays the following text:

```
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:¥Documents and Settings¥Sundaebb> program name_
```

# おわりに

つぎのステップのために

「計算の科学」副読本を参考ください