

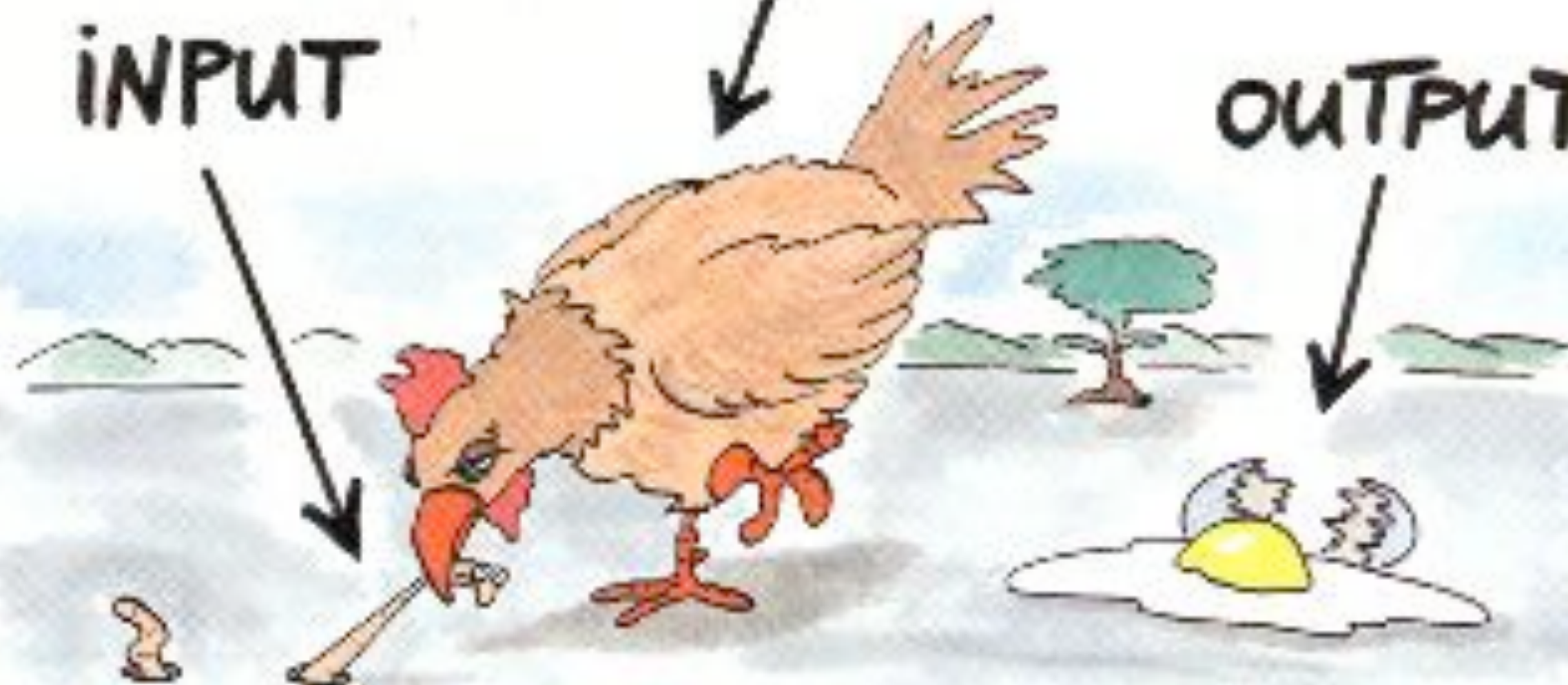
# 第4章

输入输出语句

PUT PUT

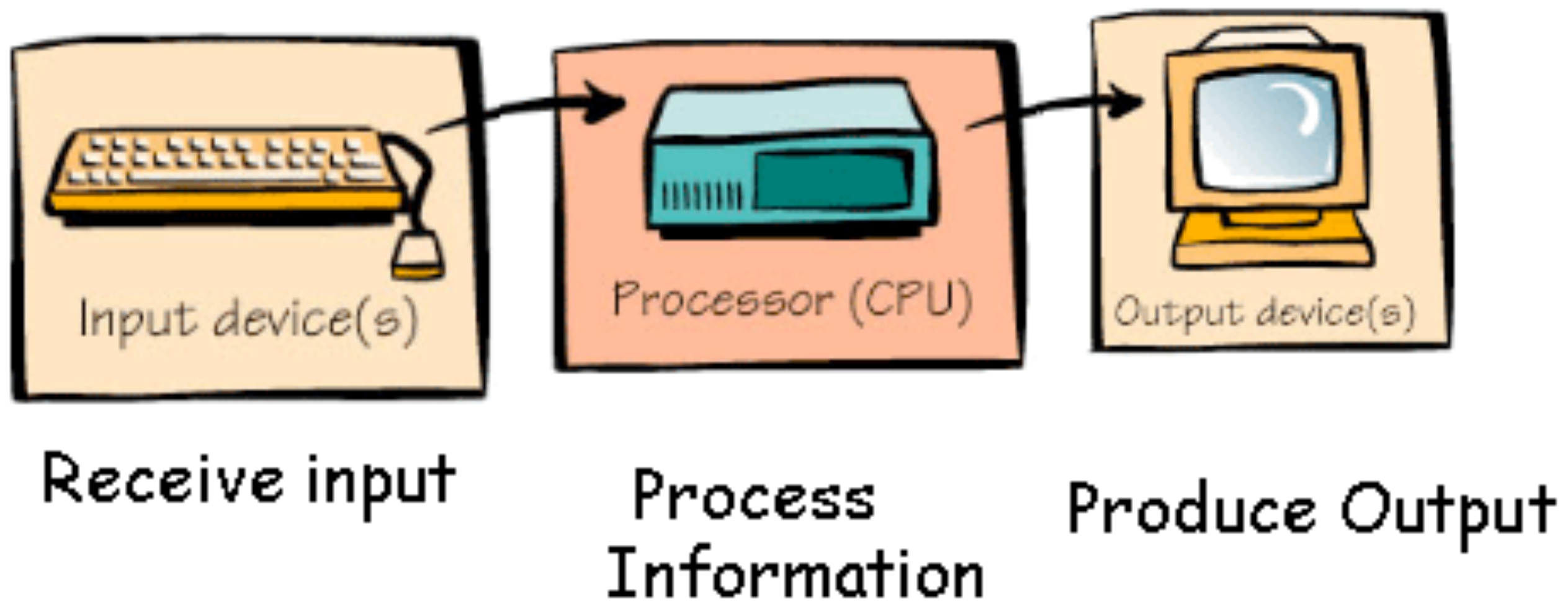
INPUT

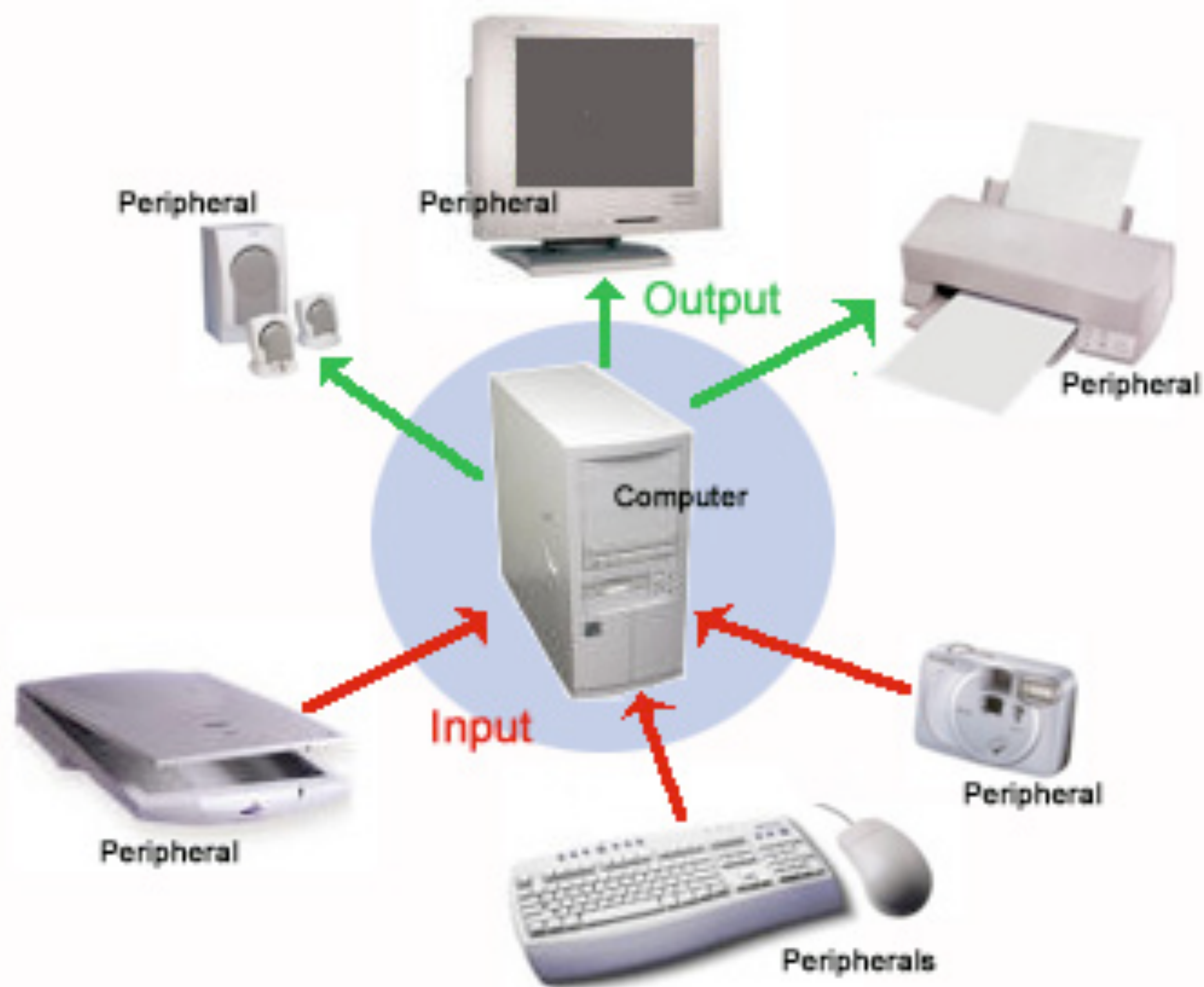
OUTPUT



@Tom

# What Computers Do





# IO库函数

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

单个字符

putchar

# putchar(<字符表达式>)

```
putchar('A');
```

```
putchar('A' + 1);
```

```
putchar(65);
```

```
putchar('\101');
```

```
putchar('\\');
```

```
putchar('\015');
```

```
putchar('\n');
```



getchar

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

```
int main()  
{  
    char c = getchar();  
    putchar(c);  
  
    return 0;  
}
```

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

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

字符串

printf

# printf

- `printf(<格式控制>, <输出列表>);`
- 输出列表: 需要输出的数据或表达式
- 格式控制: 双引号括起来的字符串
  - 普通字符
  - 格式说明: `%d`, `%f`.....

格式字符	说明
d	以带符号的十进制形式输出整型数（正数不输出符号）
o	以8进制无符号形式输出整型数（不输出前导符0）
x	以16进制无符号形式输出整型数（不输出前导符0x）
u	以无符号10进制形式输出整型数
c	以字符形式输出，只输出一个字符
s	输出字符串
f	以小数形式输出单、双精度数，隐含输出6位小数
e	以标准指数形式输出单、双精度数，数字部分小数位数为6位
g	自动选用%f或%e格式中输出宽度较短的一种格式，不输出无意义的0

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

```
int main()
```

```
{
```

```
    int a = 1, b = 2;
```

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

```
    return 0;
```

```
}
```



# 十进制整数

- `%d` 整型实际长度输出
- `%md` 输出宽度为m，右对齐
- `%-md` 输出宽度为m，左对齐
- `%ld` 输出长整型

# 八进制整数

- `%o`, `%mo`, `%-mo`, `%lo`
- 一般用于输出正整数和无符号类型数据

# 十六进制整数

- `%x`, `%mx`, `%-mx`, `%lx`
- 无负的十六进制
- 一般用于整数或无符号类型

# 无符号整数

- %u
- 输出有符号整数时，将符号位作为数值

# 字符类型

- `%c`
- 相当于 `putchar`
- `putchar`和`printf`如何选择？

# 字符串类型

- `%s` 输出字符串
- `%ms` 输出宽度为m，右对齐
- `%-ms` 输出宽度为m，左对齐
- `%.ns` 只输出字符串的前n位

# 实数

- %f
- 包括单精度float和双精度double
- %mf 输出宽度为m， 右对齐
- %.nf 保留n位小数

# 指数形式

- %e, %me, %.ne



# 较短形式浮点数

- %g
- 自动从%f和%e中选择较短的

# 注意

- 格式字母要用小写，%d不能写成%D
- 字符串中可使用转义符
- 输出%可用%%或'\45'

scanf

# scanf

- `scanf(<格式控制>, <地址列表>);`
- 格式控制: 同`printf`
- 地址列表: 数据变量的地址
  - 普通字符
  - 格式说明

# 格式字符

格式字符	说明
d	用来输入十进制整型数
o	用来输入八进制整型数
x	用来输入十六进制整型数
c	用来输入单个字符
s	用来输入字符串，以第一个分隔字符结束。
f	用来输入实型数，可以用小数形式或指数形式输入
e	与f作用相同，e与f可以互相替换

# 附加格式字符

字符	说明
l	用于输入长整数数据（可用%ld、%lo、%lx），以及double型数据
h	用于输入端整型数据（可用%hd、%ho, %hx）
m(正整数)	指定输入数据所占宽度（列数）
*	表示本输入项在读入后不赋给相应的变量

# 说明

- `scanf`不使用%u格式符，对unsigned数据，以%d、%o或%x格式输入。
- 可指定输入数据所占列数，系统自动按它截取所需数据。
- %后的附加说明符\*，表示跳过相应的数据。
- 输入数据时不能规定精度。

# scanf应注意的问题

- scanf的“格式控制”后是变量地址。
- “格式控制”中除格式说明外的其它字符原样输入。
- %c输入字符时，空格和“转义字符”都是有效输入。
- 在输入数据时，遇到以下情况视为结束；
  - 遇空格，或回车键或跳格键（tab键）。
  - 遇宽度结束时，如“%3d”，只取3列。
  - 遇非法输入。



# 程序举例

# 求三角形面积

输入三角形的变长，求三角形面积。为简单起见，设输入的三边长 $a$ ， $b$ ， $c$ 能构成三角形。

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

int main()
{
    float a, b, c, area, s;

    scanf("%f%f%f", &a, &b, &c);

    s = (a + b + c) / 2;
    area = sqrt(s * (s - a) * (s - b) * (s - c));

    printf("a = %7.2f, b = %7.2f, c = %7.2f\n", a, b, c);
    printf("area = %7.2f\n", area);

    return 0;
}
```

# 大写字母转换

从键盘输入一个大写字母，要求转换成小写字母输入。

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

```
int main()
```

```
{
```

```
    char c1, c2;
```

```
    c1 = getchar();
```

```
    printf("%c,%d\n", c1, c1);
```

```
    c2 = c1 + 32;
```

```
    printf("%c,%d\n", c2, c2);
```

```
    return 0;
```

```
}
```