

1. 十进制数可表示为 54_{10} , 而二进制数可表示为 10_2 . 其中右下标的 '10', '2' 称为基数。通常, 给定一个基数 b 和 n 个数字 $d_n d_{n-1} \dots d_1$ 组成的字符串, 该数字字符串的值可用公式计算: $\sum_{i=1}^n b^{i-1} d_i$. 如: $54_{10} = 5 * 10^1 + 4 * 10^0 = 54, 10_2 = 1 * 2^1 + 0 * 2^0 = 2$.

同时, 任意一个数可以表示成 b 以内数字的 int list 形式。如 $1100_2 = [0, 0, 1, 1], 54_{10} = [4, 5]$.

(1) 编写高阶函数: `toInt: int -> int list -> int`.

对所有 $b > 1$ 和所有 $L: \text{int list}$, 如果 L 是一个 b 进制数的 int list 表示, 函数 `toInt b L` 为其相应的整数值, `toInt b` 的结果类型为: `int list -> int`.

如: `val base2ToInt = toInt 2;`

`val 2 = base2ToInt [0, 1];`

(2) 利用数学操作 `mod` 和 `div` 可以将任意十进制整数 n 表示成基于基数 b 的 b 进制数形式, 如 $42_{10} = 132_5$.

编写高阶函数 `toBase: int -> int -> int list` 实现该转换: `toBase b n` 将十进制数 n 转换为 b 进制数的 int list 表述形式 ($b > 1, n \geq 0$).

(3) 编写高阶函数 `convert: int * int -> int list -> int list`

对任意 $b_1, b_2 > 1$ 和所有 $L: \text{int list}$ (L 为一个 b_1 进制数的 int list 表述形式), 函数 `convert(b1, b2) L` 将 b_1 进制数的 int list 表述 L 转换成 b_2 进制数的 int list 表述, 即满足 `toInt b2 (convert(b1, b2) L) = toInt b1 L`.

解答: 这个题目主要是要我们实现进制的转换, 三个函数的实现以及结果测试如下:

(1) `toInt` 函数的实现

这个函数的形式是 `toInt b L`, 主要功能是给出基数 b 以及各次幂的系数 list, 求整数 n . 即将二进制转换成 10 进制, 函数具体思路如下:

- 如果 L 为空, 则返回 0
- 如果 L 不为空, 取出 L 的第一个元素 x , 将 x 加上基数 b 乘以后面转换好的结果。而后面转换好的结果用 `toInt b L` 递归实现。

函数代码如下:

```
(* toInt: int -> int list -> int. *)
(* 给出基数b以及各次幂的系数list, 求整数n *)
fun toInt (b :int) (L :int list) :int =
  case L of [ ] => 0
  | x :: L => x + b * toInt b L
```

(2) `toBase` 函数的实现

这个函数的形式是 `toBase b n`, 将 10 进制数 n 转换为 b 进制数的 List 形式。即求 n 的 b 进制数各位的系数, 函数具体思路如下:

- 如果 n 的值是 0, 返回系数 List 位空
- 如果 n 不是 0, 将其模 b , 结果放入系数 List 的首位, 再将 n 除以 b 的值递归实现执行 `toBase` 直到最后换算完毕

函数代码如下：

```
(* toBase: int -> int -> int list *)
(* 给出基数b和整数n, 求各次幂的系数list *)
fun toBase (b :int) (n :int) :int list =
    case n of 0 => [ ]
    | _ => (n mod b) :: toBase b (n div b)
```

(3) convert 函数的实现

函数的形式是 `convert(b1, b2) L`。函数的目的是给出基数 `b1` 和 `b2` 以及基数 `b1` 的各次系数，求对应数在 `b2` 下的各次系数。

具体思路就是，先用 `toInt b1 L` 将对应的数转换为 10 进制，再用 `toBase b2 n` (`n` 是上一步的结果) 将这个数转换成以 2 位基数的系数 `List`。

函数代码如下：

```
(* convert: int * int -> int list -> int list *)
(* 给出基数b1和b2, 以及基数b1的各次幂系数, 求对应数在b2下的各次幂系数 *)
fun convert (b1 :int, b2 :int) (L :int list) :int list = toBase b2 (toInt b1 L)
```

(4) 三个函数的测试如下：

```
(* test *)
val result1 = toInt 3          (* fn : int list -> int *)
val result2 = result1 [1,1]    (* 4 *)
val result3 = toInt 3 [1,1]    (* 4 *)

val result4 = toBase 5         (* fn : int -> int list *)
val result5 = result4 28       (* [3,0,1] *)
val result6 = toBase 5 28      (* [3,0,1] *)

val result7 = convert (10,2)   (* fn : int list -> int list *)
val result8 = result7 [1,1]    (* [1,1,0,1] *)
val result9 = convert (10,2) [1,1] (* [1,1,0,1] *)

- use "hw3.sml";
[opening hw3.sml]
val toInt = fn : int -> int list -> int
val toBase = fn : int -> int -> int list
val convert = fn : int * int -> int list -> int list
val result1 = fn : int list -> int
val result2 = 4 : int
val result3 = 4 : int
val result4 = fn : int -> int list
val result5 = [3,0,1] : int list
val result6 = [3,0,1] : int list
val result7 = fn : int list -> int list
val result8 = [1,1,0,1] : int list
val result9 = [1,1,0,1] : int list
val it = () : unit
```

测试说明：

- `result1`、`result2`、`result3` 是测试 `toInt` 函数，其中 `result1` 和 `result2` 函数测试高阶函数的中间结果，`result3` 则是一次性给出全部参数。可以看出三个结果均与后面预期（注释部分）一致，函数正确。
- `result4`、`result5`、`result6` 是测试 `toBase` 函数，其中 `result4` 和 `result5` 函数测试高阶函数的中间结果，`result6` 则是一次性给出全部参数。可以看出三个结果均与后面预期（注释部分）一致，函数正确。
- `result7`、`result8`、`result9` 是测试 `convert` 函数，其中 `result7` 和 `result8` 函数测试高阶函数的中间结果，`result9` 则是一次性给出全部参数。可以看出三个结果和后面预期（注释部分）一致，函数正确。