

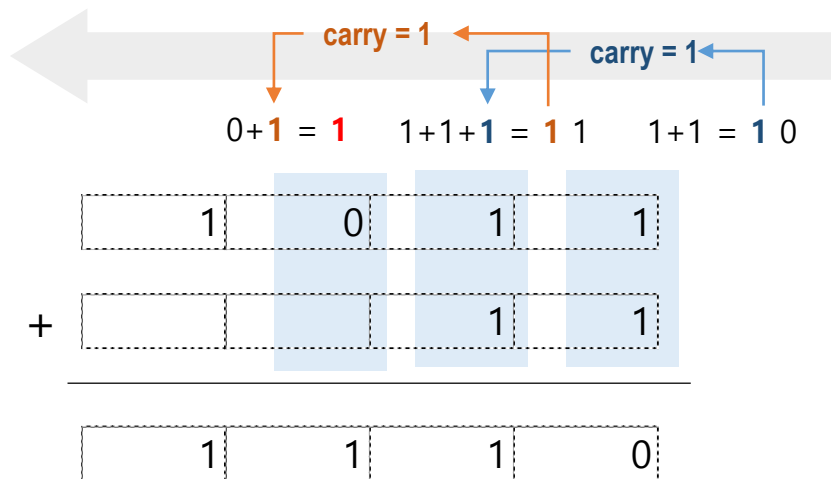
NUMBER SYSTEM (2)

– 2진수의 덧셈/뺄셈 –

Number System ^[1]

- 2 진수의 덧셈

- 10 진수를 2 진수로 변환
- $1011_{(2)} + 11_{(2)} = 1110_{(2)}$



- Carry
- Result

- Lab.01

- 파일명: bin_add.py
- using list, while-loop

- Lab.01 고려사항

- 두 입력 문자열의 bit string 을 어떻게 정수형으로 변환 할 것인가?
- 순환 방향?
 - 어느 방향으로 연산을 진행 할 것인가?
- 자리수가 다른 경우 어떻게 처리할 것인가?
- 2 진수 덧셈연산의 올림수(carry) 를 어떻게 고려할 것인가?

- 문자열 bits 로부터 정수형 리스트 변환

```
num_str_a = "1011"
num_str_b = "11"
```

```
num_bin_a = [int(x) for x in num_str_a]
num_bin_b = [int(x) for x in num_str_b]
```

- Using map

```
num_str_a = "1011"
num_str_b = "11"
```

```
num_bin_a = list(map(int, num_str_a))
num_bin_b = list(map(int, num_str_b))
```

Number System [2]

- Lab.01 고려사항

- 자리수가 다른 경우 어떻게 처리할 것인가?
 - 자리수가 큰 것 을 기준으로 순환하기로 결정.
- 2 진수 덧셈연산을 어떻게 고려할 것인가?
 - //, % 연산 사용

```
num_str_a = "1011"
num_str_b = "11"

num_bin_a = [int(x) for x in num_str_a]
num_bin_b = [int(x) for x in num_str_b]

BASE = 2
num_bin = []

bit_index = len(num_bin_a) - 1

carry = 0
pos = len(num_bin_a) - len(num_bin_b)
while bit_index >= 0:
    bit_a = int(num_bin_a[bit_index])
    if (bit_index - pos) < 0:
        bit_b = 0
    else:
        bit_b = num_bin_b[bit_index - pos]
```

```
sum_ = bit_a + bit_b + carry
carry, r = sum_ // BASE, sum_ % BASE
num_bin.insert(0, r)
bit_index -= 1
```

```
if carry:
    num_bin.insert(0, carry)
```

```
print(f"Binary number = {num_bin}")
```

- Lab.02 고려사항

- bin_add_01.py
- 자리수가 다른 경우 어떻게 처리할 것인가?
 - 자리수를 동일하게 만들고 시작하자

```
if len(num_bin_a) > len(num_bin_b):
    num_bin_b = [0] * (len(num_bin_a) - len(num_bin_b)) + num_bin_b
elif len(num_bin_a) < len(num_bin_b):
    num_bin_a = [0] * (len(num_bin_b) - len(num_bin_a)) + num_bin_a
```

Number System (3)

- Lab.02

```
num_str_a = "1011"
num_str_b = "11"

num_bin_a = [int(x) for x in num_str_a]
num_bin_b = [int(x) for x in num_str_b]

if len(num_bin_a) > len(num_bin_b):
    num_bin_b = [0] * (len(num_bin_a) - len(num_bin_b)) + num_bin_b
elif len(num_bin_a) < len(num_bin_b):
    num_bin_a = [0] * (len(num_bin_b) - len(num_bin_a)) + num_bin_a

BASE = 2
num_bin = []

bit_index = len(num_bin_a) - 1
carry = 0
while bit_index >= 0:
    bit_a = num_bin_a[bit_index]
    bit_b = num_bin_b[bit_index]
    sum_ = bit_a + bit_b + carry
    carry, r = sum_ // BASE, sum_ % BASE
    num_bin.insert(0, r)
    bit_index -= 1

if carry:
    num_bin.insert(0, carry)

print(f"Binary number = {num_bin}")
```

- Lab.03

- bin_add_02.py
- for-loop 사용

```
if len(num_bin_a) > len(num_bin_b):
    num_bin_b = [0] * (len(num_bin_a) - len(num_bin_b)) + num_bin_b
elif len(num_bin_a) < len(num_bin_b):
    num_bin_a = [0] * (len(num_bin_b) - len(num_bin_a)) + num_bin_a

BASE = 2
num_bin = []
carry = 0
for index in range(len(num_bin_a) - 1, -1, -1):
    bit_a = num_bin_a[index]
    bit_b = num_bin_b[index]
    sum_ = bit_a + bit_b + carry
    carry, r = sum_ // BASE, sum_ % BASE
    num_bin.insert(0, r)

if carry:
    num_bin.insert(0, carry)

print(f"Binary number = {num_bin}")
```

Number System ⁽⁴⁾

- HW
 - File: bin_sub_ 학번.py
 - 2 진수의 뺄셈
 - list, while-loop 사용
 - 앞의 코드 이용
 - 최종 출력 시 자리수 앞의 불필요한 0 은 제거한다.
 - Binary number =
0101010 (x)
 - Binary number =
0101010 (o)