

이재호 풀이.

### 연습문제

**응용** 1. 다음과 같이 이루어진 수열이 있다. 인자로 주어지는  $n$  번째 항까지의 합을 출력하여라.

3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, 2, 2, 2, 2, 1, 84, 2, 1, 1, 15, 3, 13, 1, 4, 2,  
6, 6, 99, 1, 2, 2, 6, 3, 5, 1, 1, 6, 8, 1, 7, 1, 2, 3, 7, 1, 2, 1, 1, 12, 1, 1, 1, 3, 1, 1, 8, 1, 1, 2,  
1, 6, 1, 1, 5, 2, 2, 3, 1, 2, 4, 4, 16, 1, 161, 45, 1, 22, 1, 2, 2, 1, 4, 1, 2, 24, 1, 2, 1, 3, 1, 2

입력: \testsum{5}

출력: 318

먼저 `clist_map_inline`을 사용해서 입력 수를 넘으면 `clist_map_break:`를 사용하여 탈출하고, 그 전에는 `clist`의 원소를 정수 표현으로 더하여 계산하는 방식입니다. 왜지 `clist_item`을 사용하여 `step` 함수를 사용하면 비효율적일 것 같아 해당 방법은 아래에 `clist_pop`과 함께 사용했습니다.

```
\ExplSyntaxOn
\clist_clear_new:N \g_testsuminput_clist
\clist_gset:Nn \g_testsuminput_clist
{
  3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, 2, 2, 2, 2, 1, 84,
  2, 1, 1, 15, 3, 13, 1, 4, 2, 6, 6, 99, 1, 2, 2, 6, 3, 5, 1, 1, 6, 8, 1,
  7, 1, 2, 3, 7, 1, 2, 1, 1, 12, 1, 1, 1, 3, 1, 1, 8, 1, 1, 2, 1, 6, 1, 1,
  5, 2, 2, 3, 1, 2, 4, 4, 16, 1, 161, 45, 1, 22, 1, 2, 2, 1, 4, 1, 2, 24,
  1, 2, 1, 3, 1, 2
}

\int_new:N \g_testsuminputcnt_int
\int_gset:Nn \g_testsuminputcnt_int { \clist_count:N \g_testsuminput_clist }

\NewDocumentCommand \testsum{ m }
{
  \int_zero:N \l_tmpa_int
  \int_zero:N \l_tmpb_int
  \clist_map_inline:Nn \g_testsuminput_clist
  {
    \int_incr:N \l_tmpa_int
    \int_compare:nTF { \l_tmpa_int <= #1 }
    {
      \int_add:Nn \l_tmpb_int { ##1 }
    }
    { \clist_map_break: }
  }
}
```

```

    \int_use:N \l_tmpb_int
  }

\testsum{5} \\
\testsum{\g_testsuminputcnt_int}
\ExplSyntaxOff

```

318  
1010

clist\_pop과 함께 이번에 새로 배운 int\_step\_inline을 활용한 방식입니다.

```

\ExplSyntaxOn
\NewDocumentCommand \testsum { m }
{
  \clist_set_eq:NN \l_tmpa_clist \g_testsuminput_clist

  \int_zero:N \l_tmpa_int
  \int_step_inline:nn { #1 }
  {
    \clist_pop:NN \l_tmpa_clist \l_tmpa_tl
    \int_add:Nn \l_tmpa_int { \l_tmpa_tl }
  }
  \int_use:N \l_tmpa_int
}

\testsum{5} \\
\testsum{\g_testsuminputcnt_int}
\ExplSyntaxOff

```

318  
1010

## 연습문제

**기본** 2. 주어진 수까지 더하는 대신 곱한다면 계승(factorial)을 구하는 trivial한 함수를 만들 수 있다. 이를 작성하여라. 단 인자는 12이하의 정수로 한다.

입력: `\simplefact{10}`

출력: 3628800

전역 변수와 별도의 cs를 만들지 않고 싶어서 `\l_tmpa_int` 스크래치 변수에 바로 곱셈을 행하였습니다. `\int_set:Nn`와 같은 함수가 없는 것 같아 `exp_args:NNo`를 사용했습니다.

```
\ExplSyntaxOn
\NewDocumentCommand \simplefact { m }
{
  \int_set:Nn \l_tmpa_int { 1 }
  \int_step_inline:nn { #1 }
  {
    \exp_args:NNo \int_set:Nn \l_tmpa_int { \l_tmpa_int * ##1 }
  }
  \int_use:N \l_tmpa_int
}
\ExplSyntaxOff

\simplefact{10}\\
\simplefact{12}
```

---

3628800  
479001600

### 연습문제

**기본** 3. 30이하의 정수 인자를 받아서  $2^n$  연산의 결과를 출력하는 함수 `\bin_power:n`을 작성하여라. 단 fp 자료형의 power 연산자 `**`를 쓰지 말고 int로 계산하여야 하며, 그 결과는 `\fp_eval:n { 2**#1 }`과 같아야 한다.

3. 입력: `\bin_power:n {8}`

출력: 256

연습문제 기본 2와 동일한 방법을 사용했습니다.

```
\ExplSyntaxOn
\cs_new:Npn \bin_power:n #1
{
  \int_set:Nn \l_tmpa_int { 1 }
  \int_step_inline:nn { #1 }
  {
    \exp_args:NNo \int_set:Nn \l_tmpa_int { \l_tmpa_int * 2 }
  }
  \int_use:N \l_tmpa_int
}

\bin_power:n {8}\\
\bin_power:n {10}
\ExplSyntaxOff
```

256

1024

## 연습문제

**실력** 4. 인자로 2진수가 주어진다. 이를 10진수로 바꾸어서 출력하는 함수를, expl3가 제공하는 진법 변환 함수를 차용하지 않고 작성하여라. 필요하다면 연습문제 3에서 작성한 `\bin_power:n`을 활용하여라.

4. 입력: `\mytoDEC{101110}`

출력: 46

들어온 값을 `tl`로 보고 문자열 조작을 한 방법입니다.

```
\ExplSyntaxOn
\NewDocumentCommand \mytoDEC { m }
{
  \tl_set:Nn \l_tmpa_tl { #1 }
  \tl_reverse:N \l_tmpa_tl

  \int_zero:N \l_tmpa_int
  \int_set:Nn \l_tmpb_int { 1 }
  \tl_map_inline:Nn \l_tmpa_tl
  {
    \int_add:Nn \l_tmpa_int { \l_tmpb_int * ##1 }
    \exp_args:NNo \int_set:Nn \l_tmpb_int { \l_tmpb_int * 2 }
  }
  \int_use:N \l_tmpa_int
}
\ExplSyntaxOff

\mytoDEC{101110} \\
\mytoDEC{100110100000101011110110}
```

46

20190710

## 연습문제

**기본** 1. 주어진 자연수를 우리말로 읽어서 그 결과를 한글로 출력하여라. 단 입력되는 숫자는 5 자리 이하로 한다.

1.

입력: \numtohangul{2019}

출력: 이천십구

한글 맞춤법 제44항에 따라 만 단위 뒤에는 띄어 쓰도록 하였습니다.

```
\ExplSyntaxOn
\tl_gset:Nn \g_prefix_tl { 일이삼사오육칠팔구 }
\tl_gset:Nn \g_suffix_tl { 십백천만 }

\NewDocumentCommand \numtohangul { m }
{
  \int_set:Nn \l_tmpa_int { \tl_count:n { #1 } - 1 }
  \tl_map_inline:nn { #1 }
  {
    \bool_if:nT
    {
      \int_compare_p:n { \l_tmpa_int > 3 } ||
      \int_compare_p:n { \l_tmpa_int == 0 } ||
      \int_compare_p:n { ##1 != 1 }
    }
    {
      \tl_item:Nn \g_prefix_tl { ##1 }
    }
    \int_compare:nT { ##1 != 0 }
    {
      \tl_item:Nn \g_suffix_tl { \l_tmpa_int }
    }
    \int_compare:nT { \l_tmpa_int == 4 } { ~ }
    \int_decr:N \l_tmpa_int
  }
}
\ExplSyntaxOff

\numtohangul{2019} \\
\numtohangul{11019} \\
\numtohangul{909}
```

이천십구

일만 천십구

구백구

## 연습문제

**실력** 2. 두 수를 인자로 받아 그 최대공약수를 출력하는 명령 \mygcd를 작성하여라.

2.

입력: \mygcd{16}{24}

출력: 8

Python 코드를 직역하였습니다.

```
\ExplSyntaxOn
\int_new:N \l_tmpc_int
\NewDocumentCommand \mygcd { m m }
{
  \int_compare:nTF { #1 > #2 }
  {
    \int_set:Nn \l_tmpa_int { #1 }
    \int_set:Nn \l_tmpb_int { #2 }
  }
  {
    \int_set:Nn \l_tmpa_int { #2 }
    \int_set:Nn \l_tmpb_int { #1 }
  }

  \int_while_do:nn { \l_tmpb_int != 0 }
  {
    \int_set:Nn \l_tmpc_int
    {
      \int_mod:nn { \l_tmpa_int } { \l_tmpb_int }
    }
    \int_set:Nn \l_tmpa_int { \l_tmpb_int }
    \int_set:Nn \l_tmpb_int { \l_tmpc_int }
  }

  \int_use:N \l_tmpa_int
}
\ExplSyntaxOff

\mygcd{16}{24}\
\mygcd{19077743}{15973807}
```

8

4409