TEX と言えば数式です。例えば、次の様な数式が綺麗に表現できます。

(TEX **の表現**)

```
\[
\sum_{i=1}^{n} a_i = a_1 + \cdots + a_i + \cdots + a_n
\]
```

(TypeSet **の結果**)

$$\sum_{i=1}^{n} a_i = a_1 + \cdots + a_i + \cdots + a_n$$

次に、これと良く「似」た、別の数式を考えます。

(TEX **の表現**)

```
\[
\sum_{j=1}^{n} a_j = a_1 + \cdots + a_j + \cdots + a_n
\]
```

(TypeSet **の結果**)

$$\sum_{j=1}^{n} a_j = a_1 + \cdots + a_j + \cdots + a_n$$

これは、添字を表す変数文字が i から j に変化したものです。もちろん、同様にして i を k に換えたものも考えられますが、この様に「文字列の一部だけを変更して再利用したい」場合があります。

この様な場合は、「引数付きマクロ」を定義して、利用すると便利です。

(**引数付マクロ** SumX **の定義**)

引数 : 「変更したい部分」を「後から (引数として) 指定できる」様にする仕組

引数は、複数指定できるが、順番に #1, #2, .. で「引数の値を参照」できる

```
\newcommand{\SumX}[1]{%
\sum_{#1=1}^{n} a_{#1} = a_1 + \cdots + a_{#1} + \cdots + a_n%
}
```

これを利用すれば、簡単に、一部を差し替えた式が表現できます。

(TEX **の表現**)

```
\[
\SumX{i}=\SumX{j}
\]
```

(TypeSet **の結果**)

$$\sum_{i=1}^{n} a_i = a_1 + \cdots + a_i + \cdots + a_n = \sum_{j=1}^{n} a_j = a_1 + \cdots + a_j + \cdots + a_n$$

ポイントは三つあります。その内、二つは、マクロ定義部分にあり、その内の一つは [1] の部分で、これは「このマクロは一つの引数を持つ」という意味です。もう一つは、マクロ定義の内容の中の#1 の部分で、これ

は、「ここが引数の内容に置き換わる」という意味です。そして、最後の一つは、マクロを利用する時に、マクロ名の後ろに追加された引数です。マクロが展開されるときに、マクロの内容に含まれている`#1` がこの引数に差し替えられます。

もちろん、この引数は、複数指定できる様にもできます。例えば、次の例は、更に、数列の項目名や上限まで変更できる版 (なので三つの引数を持つ..) です。

(引数付マクロ `SumXAN` の定義)

```
\newcommand{\SumXAN}[3]{%
\sum_{#1=1}^{#3} #2_{#1} = #2_1 + \cdots + #2_{#1} + \cdots + #2_{#3}%
}
```

(TEX の表現)

```
\[
\SumXAN{k}{b}{m}
\]
```

(TypeSet の結果)

$$\sum_{k=1}^{m} b_k = b_1 + \cdots + b_k + \cdots + b_m$$

もしかしたら、初項が 1 でなく 0 にしたい場合もあるかもしれません。この様な場合にも対応可能な、四つの引数を持つマクロも考えてみましょう。