

*** Use of link module library ***

1. link module library

The link module library is used to achieve a high-speed link list in a descartes language.

It is possible to treat as well as an interactive list of a double link. Moreover, an arbitrary, structural data can be added as an element of link.

They are the following procedures when using it.

1) Include does the link module.

```
? <include link>;
```

2) The object that inherits to the link module is made.

```
::<lk
```

```
inherit link;
```

```
>;
```

In the example above, the lk object is newly defined and the link module has been inherited to.

3) The object newly made is scanned in both direction before and after the addition, the deletion, and the list of the node.

2. Usage example

Let's show the usage example below.

```
? <include link>;
```

```
::<lk
```

```
inherit link;
```

```
>;
```

```
? ::lk <getlist #l>;
```

```
? ::lk <length #len>;
```

```
? ::lk <put a>;
```

```
? ::lk <put b>;
```

```
? ::lk <put c>;
```

```
? ::lk <put d>;
```

```
? ::lk <put e>;
```

```
? ::lk <put f>;
```

```
? ::lk <getlist #l>;
```

```
? ::lk <middle_node #n>;
```

```
? <list>;
```

```
? ::lk <next #p>;
```

```

? ::lk <prev #p>;
? ::lk <prev>;
? ::lk <prev #p>;
? ::lk <next>;
? ::lk <prev #p>;
? ::lk <prev #p>;
? ::lk <prev #p>;
? ::lk <prev #p>;

? ::lk <get #v>;

? ::lk <getlist #l>;

? <list>;

? ::lk <length #n>;

? ::lk <setlist (1 2 3 4 5 6 7 8 9 10)>
  ::lk <getlist _>
  ::lk <length #n>;

? ::lk <next _>
  ::lk <next _>
  ::lk <del>
  ::lk <getlist _>;

? ::lk <del>;
? ::lk <getlist _>;

? ::lk <top _>;
? ::lk <last _>;
? ::lk <top _>;
? ::lk <last _>;
? ::lk <getlist _>;

```

3. Operation of link

The following methods can be used.

<prev #prev1>

It moves to previous one link.

<next #next1>

It moves to link in the one back.

<clear>

All elements of link are deleted and cleared.

<put #v>

The value set to the argument is added as one node element of link.

<get #v>

The node element of link being indicated now is taken out to the argument.

The node element of link being indicated now is deleted.

<getlist #list>

The list that connects all elements connected with link is returned.

<setlist #list>

The list set to the argument is registered as link.

<length #len>

The length of the node element connected with link is returned.

<top #n>

It moves to the head of link.

<last #n>

It moves to the tail of link.