DRAFT PROPOSAL

CppNorth 2023

Giving Key a Vocabulary

Adám Brudzewsky

Key is awesome

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      {α}⌸names
Blake
Angel
Emery
Devon
```

Key is awesome

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      {⍺ ⍵}⌸names
Blake  1 7
Angel  2 4 8 9
Emery  3
Devon  5 6
```

Key is awesome

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      {⍺ ⍵ (≠⍵)}⌸names
Blake  1 7       2
Angel  2 4 8 9   4
Emery  3         1
Devon  5 6       2
```

Key: sort?

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      {⍺ ⍵ (≠⍵)}⌸names
Blake  1 7     2
Angel  2 4 8 9 4
Emery  3       1
Devon  5 6     2
```

Key: sort

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      {⍵[⍋⍵;]} {⍺ ⍵ (≠⍵)}⌸names
Angel  2 4 8 9  4
Blake  1 7      2
Devon  5 6      2
Emery  3        1
```

Key: add items?

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      {⍵[⍋⍵;]} {⍺ ⍵ (≠⍵)}⌸names
Angel   2 4 8 9   4
Blake   1 7       2
Colby             0
Devon   5 6       2
Emery   3         1
```

Key: add items

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De
      all←↑'Angel' 'Blake' 'Colby' 'Devon' 'Emer
      {⍵[⍋⍵;]} {i←¯1↓⍵ ⋄ ⍺ i (≠i)}⌸names,all
Angel  2 4 8 9  4
Blake  1 7      2
Colby           0
Devon  5 6      2
Emery  3        1
```

Key: add items, order

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De
      all←↑'Angel' 'Blake' 'Colby' 'Devon' 'Emer
      {i←(1↓⍵)-≠all ⋄ ⍺ i (≠i)}⌸all;names
Angel  2 4 8 9  4
Blake  1 7      2
Colby           0
Devon  5 6      2
Emery  3        1
```

Key: add items, order, query?

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De
      all←↑'Colby' 'Emery'
      {i←(1↓⍵)-≠all ⋄ ⍺ i (≠i)}⌸all,names
Colby          0
Emery  3       1
Blake  7       1
Angel  4 8 9   3
Devon  6       1
```

Key: add items, order, query

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De
      all←↑'Colby' 'Emery'
      (≠all)↑{i←(1↓⍵)-≠all ◇ ⍺ i (≠i)}⌸all,names
Colby          0
Emery  3       1
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (2={≠⍵}⌸names)⌿∪names
Blake
Devon
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (2={≠⍵}⌸names)⌿∪names
Blake
Devon
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

          {≠⍵}⌸names
2 4 1 2
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      {⍺ (≠⍵)}⌸names
Blake  2
Angel  4
Emery  1
Devon  2
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      ⍀{⍺ (≠⍵)}⌸names
Blake  Angel  Emery  Devon
    2      4      1      2
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      ↓⌸{α (≠⍵)}⌸names
Blake  Angel  Emery  Devon   2 4 1 2
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (k c)←↓⍉⌸{⍺ (≠⍵)}⍸names
      ↑(2=c)≠k
Blake
Devon
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      ⊃{↑(2=⍵)⌿⍺}⌿ ↓⍉⍪{⍺ (≠⍵)}⌸names
Blake
Devon
```

Key: get sorted keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      ⊃{↑(2=⍵)⌿⍺}/ ↓⍉ {⍵[⍒⍵;]} {⍺ (≠⍵)}⌸names
Devon
Blake
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (2={≠⍵}⌸names)⌿∪names
Blake
Devon
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (2={≠⍵}⌸names)⌿voc←∪names
Blake
Devon
```

Key: get keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (2={≠⍵}⌸voc⊢names)⌿voc←∪names
Blake
Devon
```

Key: get sorted keys

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (2={≠⍵}⌸voc⊢names)≠voc←{⍵[⍒⍵;]}∪names
Devon
Blake
```

Key: get sorted keys

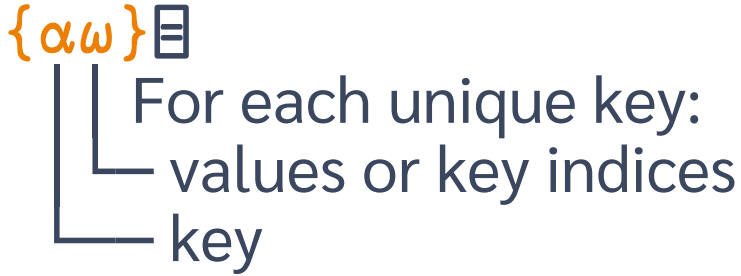
```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De

      (2={≢⍵}⌸voc⊢names)≠voc←{⍵[⍋⍵;]}∪names
Devon
Blake
```

What ***can*** we do with that operand?

$\{\neq\omega\}$

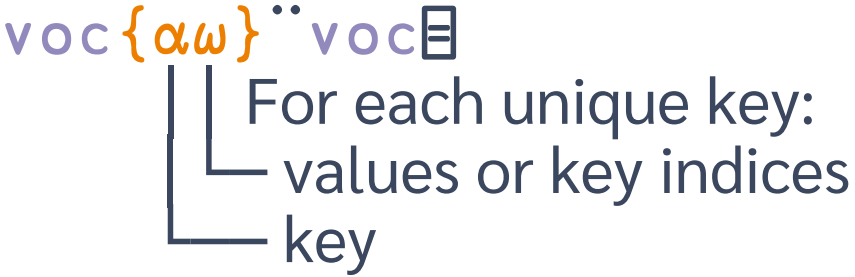
What ***can*** we do with that operand?



What ***can*** we do with that operand?

{αω}/voc⊟
- For each unique key:
  - values or key indices
- key

What ***can*** we do with that operand?



What ***can*** we do with that operand?

```
{≠ω}¨voc⌸
    │ For each unique key:
    └─ values or key indices
```

What ***can*** we do with that operand?

Key: get keys

```
names←'Blake' 'Angel' 'Emery' 'Angel' 'Dev

(2={≠⍵}⌸names)≠∪names
```

Blake	Devon

```
≠⍤voc⌸
```

Key proposal: get keys

```
names←'Blake' 'Angel' 'Emery' 'Angel' 'Dev
```

```
(2=≢⌸names)≠∪names
```

Blake	Devon

```
(2=≢¨voc⌸names)≠voc←∪names
```

Blake	Devon

Key proposal: get sorted keys

```
names←'Blake' 'Angel' 'Emery' 'Angel' 'Dev
```

```
{ω[⍒ω]}(2={≢ω}⌸names)⌿∪names
```

Devon	Blake

```
(2=≢¨voc⌸names)⌿voc←∪names
```

Blake	Devon

Key proposal: get sorted keys

```
names←'Blake' 'Angel' 'Emery' 'Angel' 'Dev
```

```
{ω[⍒ω]}(2={≠ω}⌸names)/∪names
```

Devon	Blake

```
(2=≠⌸voc⌸names)/voc←{ω[⍒ω]}∪names
```

Devon	Blake

Key proposal: query

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De
      all←↑'Colby' 'Emery'
      (≠all)↑{i←(1↓⍵)-≠all ◇ ⍺ i (≠i)}⌸all,names
Colby      0
Emery  3   1
```

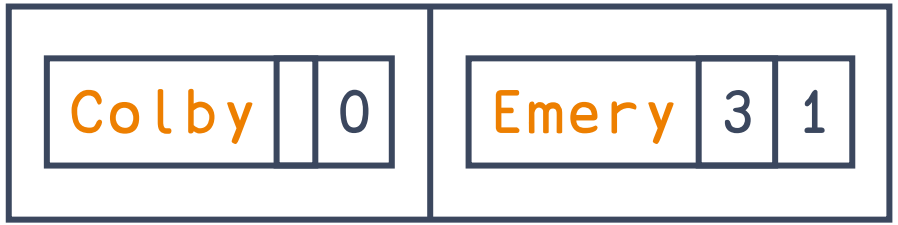
Key proposal: query

```
      names←↑'Blake' 'Angel' 'Emery' 'Angel' 'De
      all←↑'Colby' 'Emery'
      (≠all)↑{i←(1↓⍵)-≠all ◇ ⍺ i (≠i)}⌸all,names
Colby      0
Emery   3  1

      ↑(↓all){⍺ ⍵ (≠⍵)}⌸all⌸names
Colby      0
Emery   3  1
```

Key proposal: query

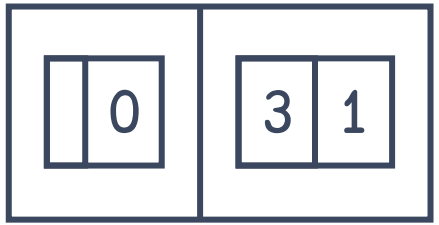
```
names←'Blake' 'Angel' 'Emery' 'Angel' 'Dev
all←'Colby' 'Emery'
(≠all)↑{i←(1↓⍵)-≠all ◇ ⊂⍺ i (≠i)}⌸all,name
```



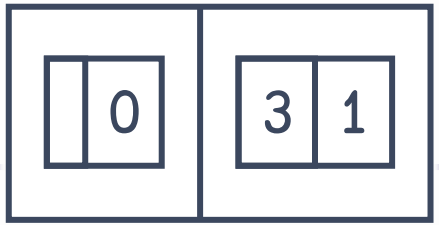

```
all{⍺ ⍵ (≠⍵)}⌸all⌸names
```

Key proposal: query

```
names←'Blake' 'Angel' 'Emery' 'Angel' 'Dev
all←'Colby' 'Emery'
(≠all)↑{i←(1↓⍵)-≠all ◇ ⊂i (≠i)}⌸all⍪names
```




```
{⍵ (≠⍵)}¨all⌸names
```



Giving Key a Vocabulary

Function Operand	Array Operand
`Fm⍢ vals`	`voc⍢ vals`
`Fd⍢ vals`	`voc⍢ vals`
`keys Fm⍢ vals`	`keys(voc⍢)vals`
`keys Fd⍢ vals`	`keys(voc⍢)vals`

Giving Key a Vocabulary

DRAFT PROPOSAL

Function Operand

```
      Fm⌸ vals
      Fd⌸ vals
keys  Fm⌸ vals
keys  Fd⌸ vals
```

Array Operand

```
      Fm⍤⍤voc⌸ vals
voc   Fd⍤⍤voc⌸ vals
      Fm⍤⍤keys(voc⌸)vals
voc   Fd⍤⍤keys(voc⌸)vals
```

```
]get is.gd/KeyVoc:aplo
```