Lexographically → Increasing order.

String Sort → Sort according to the Comparison function!

```
void String_Sort (const char **arr, const int cnt, int (*cmp_func)(const char *a,
                                                    const char *b)){
}
```

arr → array of string

cnt → length of string array.

cmp_fun → function pointer.

---

int lexicographic_Sort (char *, char *) → non decending order

int lexicographic_Sort_reverse (char *, char *) → lexicographically non-increasing order.

int Sort_by_numof_distinct_char (char *, char *) → non decreasing order (characters) — lexicographically

int Sort_by_length (char *, char *) → Same length → lexicograph. Smaller first! (distinct chars).

String: lower case

wkue
nop
sbr
felis

→ length →

nop
sbr
wkue
felis.

a, r, s, t, w

distinct characters

26 characters → Same number → non decreasing order

lexicographic sort

Non decreasing.

> → 1
< → -1
= → 0

lexicograph reverse

non increasing order

> → -1
< → 1
= → 0.

a > b → +ve
a < b → -ve
a == b → 0

word! length

④

Non decreasing order.

bigger → 1
smaller → -1
Same → 0.

Sort gek →

length 4

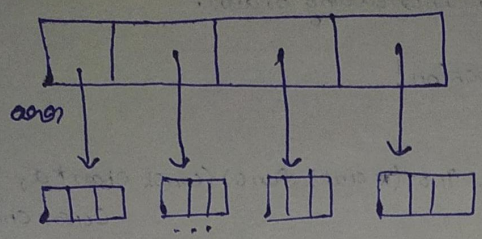
① felis nop sbr wkue.

nop felis sbr wkue

② nop sbr felis wkue

③ nop sbr wkue felis.

a > b
Swap
else
Leave



Swap

① dereference go to pointer
touch that pointer
change it value
(stored address)

---

ab, bc, cd → Permutation: $3! = 3 \times 2 = 6$

ab, ab, bc → $\frac{3!}{2!} = \frac{3 \times 2}{2} = 3$.

ab	ab	bc
ab	bc	ac
bc	ab	ab

* $a[k] < a[k+1]$ → find k [No such → last permutation] [largest index k]
* largest index l greater than k $a[k] < a[l]$
* Swap $a[k]$ and $a[l]$
* Reverse sequence from $a[k+1]$ up to and including final element.

[1, 2, 3, 4]

1. $k = 2$, $a[2] < a[3]$ | $k = 2$
2. 4 is the only value largest than 3 ($l = 3$) index
3. Swap 1 2 4 3
4. Reverse sequence: 1 2 4 3
   from k+1 to n

---

① Find k

1 2 3 4

index	
0	1 < 2
1	2 < 3
2	3 < 4

index
k = 2

worcase: sorted – don't worry!

0 to n-1 (excluding)
if $a[i] < a[i+1]$
k = i;

Finally we get highest k possible (index)

Find l

$a[k] < a[l]$

< = → No

∴ array[k] < array[k+1]

Always exist!

```
for (i=0; i<n; i++)
{
  if (array[k] < array[i])
      l = i;
}
return l;
```

③ Swap a[k] and a[l].

Swap function

Reverse sequence

from a[k+1] up to n

even:

1 2 3 4 5 6

6 5 4 3 2 1

Run: $\frac{6}{2} - 1 = 2$ (index)

odd:

1 2 3 4 5

Leave!

$\frac{5}{2} - 1 = 1$ (index)

If between

1 2 3 4 5 6

1 2 6 5 4 3

Start:

$\frac{3+6}{2} = 1^{\bullet} = \frac{9}{2} - 1$

$= 4 - 1$

$= 3$ (element)!

index

(go up to index 3)

1 2 3 4 5 6 7

$\frac{3+7}{2} - 1 = \frac{10}{2} - 1$

$= 5 - 1$

$= 4.$

Swap:

index:

6, 2

5, 3

4, 4

Start → Start index + 1

end → end index + 1

Go more generic (index)

1 2 3 4 5 6

start index: 2
end index: 5

$\frac{2+5}{2} = 3$

1 2 6 5 4 3

1 2 3 4 5 6 7

$\frac{2+6}{2} = 4$ (go up to 5)

1 2 7 6 5 4 3

Reverse from K+1 to n

4! = 4×3×2

working - for numbers!

1, 2, 2 → $\frac{3!}{2!} = 3$

Lexographically! (given)!

→ "ab", "ab", "bc"

→ ab bc ab

0, 1

1 1 2

1 < 1 < 2

1 2 1

0 < 2

1 2 1

1 2 1

1 < 2

yes

K = 1

l = 2

1 1 2

1 2 1

1 2

2 1 1

K = 0

l = 1

array[i] < array[i+1]