SLICING DEMYSTIFIED

index	0	1	2	3	4	5	6	7	8	9	10
character	**'H'**	**'E'**	**'L'**	**'L'**	**'O'**	**' '**	**'W'**	**'O'**	**'R'**	**'L'**	**'D'**
index	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

- Extract part of a string using the syntax **[start:stop:step]** which extract characters from index **start** until index **stop** (exclusive) in increments of **steps**.
- Omitted index: if we omit **start**, it defaults to (first) index (i.e., zero). If we omit **stop**, it defaults to (last) index in the sequence. If we omit **step**, it defaults to (1).
- Slicing produce copies of the original string

Given
S=”HELLO WORLD”

Examples

[start:stop:step]

```
print(S[0:5:1])            # HELLO
print(S[6:11:1])           # WORLD

print(S[0:11:1])           # HELLO WORLD
print(S[0:11:2])           # HLOWRD   index:0,2,4,10
print(S[0:11:3])           # HLWL   index:0,3,6,9

print(S[-11:-1:1])         # HELLO WORL

print(S[-1:-12:-1])        # DLROW OLLEH  index: -1,-2,-3,…,-10,-11
print(S[-1:-12:-2])        # DRWOLH   index: -1,-3,-5,-7,-9,-11
print(S[-1:-13:-2])        # DRWOLH   index: -1,-3,-5,-7,-9,-11
```

[start:stop] omitted step defaults to 1

```
print(S[6:12])          # S[6:12:1]: WORLD
```

[:stop:step] omitted start defaults to 0

```
print(S[:5:1])          # S[0:5:1]:  HELLO
```

[start: :step] omitted stop defaults to last index in sequence
Write down the sequence to determine the 'last' index in sequence:

```
print(S[0::1])       # index 0,1,2,...11:  HELLO WORLD
print(S[0::2])       # index 0,2,4,6,8,10:  HLOWRD
print(S[0::3])       # index 0,3,6,9:  HLWL
# notice the last index above is different depending on step
```

[: :step] omitted start,stop

```
print(S[::3])        # index 0,3,6,9:  HLWL
```

[: :] omitted start,stop,step

```
print(S[::])         # index 0,1,2,...,10:  HELLO WORLD
print(S[:])          # index 0,1,2,...,10:  HELLO WORLD
print(S)             # HELLO WORLD  --this is not a slice--
# first two are copies of S, last one is the real S
```