

QUIZ 2

COMP9021 PRINCIPLES OF PROGRAMMING

```
$ python3 quiz_2.py
```

```
Enter two positive integers (possibly 0): 0 6
```

```
Here is the list L of generated values:
```

```
[3, 3, 0, 2, 4, 3]
```

```
Removing again and again in a copy of L the values  
with strictly larger neighbours (on both sides):
```

```
[3, 3, 4, 3]
```

```
Dictionary mapping i to (j1, j2), with j1 minimal and j2 maximal,  
such that for all j with j1 <= j <= j2, the value of L at location j  
is at least equal to the value of L at location i:
```

```
{0: (0, 1), 1: (0, 1), 2: (0, 5), 3: (3, 5), 4: (4, 4), 5: (4, 5)}
```

```
$ python3 quiz_2.py
```

```
Enter two positive integers (possibly 0): 0 8
```

```
Here is the list L of generated values:
```

```
[6, 6, 0, 4, 7, 6, 4, 7]
```

```
Removing again and again in a copy of L the values  
with strictly larger neighbours (on both sides):
```

```
[6, 6, 7, 7]
```

```
Dictionary mapping i to (j1, j2), with j1 minimal and j2 maximal,  
such that for all j with j1 <= j <= j2, the value of L at location j  
is at least equal to the value of L at location i:
```

```
{0: (0, 1), 1: (0, 1), 2: (0, 7), 3: (3, 7), 4: (4, 4),  
5: (4, 5), 6: (3, 7), 7: (7, 7)}
```

```
$ python3 quiz_2.py
```

```
Enter two positive integers (possibly 0): 1 10
```

```
Here is the list L of generated values:
```

```
[2, 9, 1, 4, 1, 7, 7, 7, 6, 3]
```

```
Removing again and again in a copy of L the values  
with strictly larger neighbours (on both sides):
```

```
[2, 9, 7, 7, 7, 6, 3]
```

```
Dictionary mapping i to (j1, j2), with j1 minimal and j2 maximal,  
such that for all j with j1 <= j <= j2, the value of L at location j  
is at least equal to the value of L at location i:
```

```
{0: (0, 1), 1: (1, 1), 2: (0, 9), 3: (3, 3), 4: (0, 9),  
 5: (5, 7), 6: (5, 7), 7: (5, 7), 8: (5, 8), 9: (5, 9)}
```

```
$ python3 quiz_2.py
```

```
Enter two positive integers (possibly 0): 3 12
```

```
Here is the list L of generated values:
```

```
[3, 9, 8, 2, 5, 9, 7, 10, 9, 1, 9, 0]
```

```
Removing again and again in a copy of L the values  
with strictly larger neighbours (on both sides):
```

```
[3, 9, 9, 10, 9, 9, 0]
```

```
Dictionary mapping i to (j1, j2), with j1 minimal and j2 maximal,  
such that for all j with j1 <= j <= j2, the value of L at location j  
is at least equal to the value of L at location i:
```

```
{0: (0, 2), 1: (1, 1), 2: (1, 2), 3: (0, 8), 4: (4, 8), 5: (5, 5),  
 6: (5, 8), 7: (7, 7), 8: (7, 8), 9: (0, 10), 10: (10, 10), 11: (0, 11)}
```