Task 1

Create this pattern:

```
      pattern
zzzzzzHHHHzzzzzzHHHH
HHHHzzzzzzHHHHzzzzzz
zzzzzzHHHHzzzzzzHHHH
HHHHzzzzzzHHHHzzzzzz
```

Try to make your code as short as you can!

Put the expression into your submission:

```
pattern ← …
```

Task 2

Rewrite the following expressions to not use parentheses ( ):

```
a: (÷a)×b
b: (÷a)÷b
c: (a+b)-5
d: (a+b)+5
```

Tip: assign different values to `a` and `b` to test your solutions

Put the expressions into your submission:

```
a: b×÷a
b:
c:
d:
```

Task 3

Define a function `ID` to return an ω by ω identity matrix:

```
      ID 3
1 0 0
0 1 0
0 0 1
      ID 5
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

Put the function definition into your submission:

`ID ← {…}`

Task 4

In mathematics, there is a function called "factorial".
The factorial of 1 is 1:

```
      Factorial 1
1
```

The factorial of any other number n is the number times the factorial of the previous number.

```
      Factorial 5
120
      Factorial 4
24
      5 × Factorial 4
120
      Factorial 13
6227020800
```

Write a Factorial function.

Put the function definition into your submission:

```
Factorial ← {…}
```

Task 5

Define a function `TimesRows` to multiply a matrix left argument and a vector right argument. For example:

```
      mat ← 4 3⍴6 9 9 10 9 1 3 5 3 6 7 3
      vec ← 1 0 ¯1

      mat TimesRows vec
 6 0 ¯9
10 0 ¯1
 3 0 ¯3
 6 0 ¯3

      (2 3⍴⍳6) TimesRows 10*¯1+⍳3
1 20 300
4 50 600
```

Put the function definition into your submission:

`TimesRows ← {…}`

Task 6

In a school's science Olympiad (a problem-solving competition), students are given a "Gold", "Silver" or "Bronze" award based on their performance in a written test. Students who score above 85 receive Gold; those who score above 70 get Silver, while those who score over 40 get Bronze. Write a function to convert numeric scores into the letters G, S or B for Gold, Silver or Bronze, or a hyphen (-) for no award.

```
      Award 30 49 92 90 34 94 59 41 89
GS--G-SS-
      Award 79 27 58 52 90
BGSS-
```

Put the function definition into your submission:

`Award ← {…}`

Task 7

The 3D array `rain` gives the monthly rainfall in millimetres over 7 years for 5 countries.

```
⎕RL←42
rain ←?7 5 12⍴250
```

Write a function to find the average monthly rainfall for each individual month in each of the 5 countries.

```
      ⌊ MonthAvg rain
117 137 125 106 130 133 172 118  91 140 133 113
116 146 102 147 105  73 111 138 158 128 144 126
124 106 126 101 172 126 182 109 174 126  59 135
109 121 192 138 100 131  68 156 123 140 110 159
121 120 138 147  75 132 111 102 118 117 157 109
```

Put the function definition into your submission:

`MonthAvg ← {…}`

Task 8

Assign scalar numeric values (single numbers) to the variables `years`, `countries` and `months` such that the rain data can be summarised as follows:

```
      ⍴+/[years]rain        ⍝ Sum over years
5 12
      ⍴+/[countries]rain    ⍝ Sum over countries
7 12
      ⍴⌈/[months]rain       ⍝ Max over month
7 5
```

Put the expressions into your submission:

```
years ← …
countries ← …
months ← …
```

Task 9

Define a function `RemoveEvery` to remove every ω^{th} number from the vector α:

```
      (⍳10) RemoveEvery 4
1 2 3 5 6 7 9 10
      (⍳15) RemoveEvery 3
1 2 4 5 7 8 10 11 13 14
      6 2 9 4 1 2 6 RemoveEvery 2
6 9 1 6
      'hello world' RemoveEvery 3
helowold
      □A RemoveEvery 5
ABCDFGHIKLMNPQRSUVWXZ
```

Put the function definition into your submission:

`RemoveEvery ← {…}`

Task 10

Bus stops in a town are labelled A to E. Define a function `RouteMatrix` which returns a Boolean matrix where 1s indicate that buses go from one bus stop to the next.

For example, for routes given in the tables below:

```
      RouteMatrix 'BE' 'C' 'AE' 'BCE' 'A'
0 0 1 0 1
1 0 0 1 0
0 1 0 1 0
0 0 0 0 0
1 0 1 1 0
```

Depart from	Arrive at
A	B and E
B	C
C	A and E
D	B, C and E
E	A

```
      'ABCDE'⍪RouteMatrix 'C' 'CDE' 'ABDE' 'E' 'B'
A B C D E
0 0 1 0 0
0 0 1 0 1
1 1 0 0 0
0 1 1 0 0
0 1 1 1 0
```

Depart from	Arrive at
A	C
B	C, D and E
C	A, B, D and E
D	E
E	B

Put the function definition into your submission:

`RouteMatrix ← {…}`

Task 11

Write a function `OverUnder` that takes:
- A 3-element vector left argument `⍺`
- Any simple numeric array right argument `⍵`

The function returns:
- An array with the same shape as `⍵` in which:
  - If a number is strictly greater than `⍺[1]`, it becomes `⍺[2]`
  - Otherwise, it becomes `⍺[3]`

```
      12 29 5 OverUnder 3 ¯4 15 0 19 5
5 5 29 5 29 5

      0 5 ¯6 OverUnder 3 5⍴¯3 0 5 1 4 ¯1 5 3 ¯5
¯6 ¯6 5  5  5
¯6  5 5 ¯6 ¯6
¯6  5 5  5 ¯6

      15  1 0 OverUnder 2 2 7⍴11 14 17 14 13 19 11 16 17 15 16 15 19
0 0 1 0 0 1 0
1 1 0 1 0 1 0

0 1 0 0 1 0 1
1 0 1 0 1 0 0
```

Put the function definition into your submission:

`OverUnder ← {…}`