

2013 International APL Problem Solving Competition - Phase I

Phase I Tips

We have provided you with several test cases for each problem to help you validate your solution.

We recommend that you build your solution using dfns. A dfn (direct function) is one or more APL statements enclosed in braces {}.

The left hand argument, if any, is represented in a dfn by α , while the right hand argument is represented by ω .

Example:

```
'Hello' { $\alpha$ , '-',  $\omega$ , '!'} 'world'  
Hello-world!
```

The result of a dfn is the value of the first result producing statement.

Example:

```
'left' {  $\omega$   $\diamond$   $\alpha$  } 'right'  
right
```

For more information on dfns you can refer to page 152 in Mastering Dyalog or use the online help included with Dyalog APL.

The symbol $\#$ is the APL comment symbol. In some of the examples below, comments are provided to give more information.

Phase I Problems:

Sample Problem - I'd like to buy a vowel

Write a dfn to count the number of vowels in a character vector.

When passed the character vector 'APL Is Cool', your solution should return:

4

Below are 2 sample solutions. Both produce the correct answer, however the first solution would be ranked higher by the competition judging committee as it demonstrates better use of array oriented programming.

```

    {+/ω∈'AEIOUaeiou'}'APL Is Cool' ⌞ better solution
⌞
    {(+/ω='A')+(/ω='E')+(/ω='I')+(/ω='O')+(/ω='U')+(/ω='a'))
+(/ω='e')+(/ω='i')+(/ω='o')+(/ω='u'))}'APL Is Cool' ⌞ lesser
solution
⌞

```

Problem 1 - Seems a bit odd to me

Write a dfn to produce a vector of the first n odd numbers.

Test cases:

```

    {your_solution} 10
1 3 5 7 9 11 13 15 17 19

    {your_solution} 1
1

    {your_solution} 0 ⌞ this should return an empty vector

```

Problem 2 - Making the grade

Write a dfn which returns the percent (from 0 to 100) of passing (65 or higher) grades in a vector of grades.

Test cases:

```

    {your_solution} 25 90 100 64 65
60

    {your_solution} 50
0

    {your_solution} 80 90 100
100

    {your_solution} 10 ⌞ all grades in an empty vector are passing
100

```

Problem 3 - What's in a word

Write a dfn which returns the number of words character vector.

For simplicity's sake, you can consider the space character ' ' to be the only word separator.

Test cases:

```
{your_solution} 'Testing one, two, three'
```

4

```
{your_solution} '' A empty vector has no words
```

0

```
{your_solution} ' this vector has extra blanks ' A just  
counting the blanks won't work
```

5

Problem 4 - Keeping things in balance

Write an APL dfn which returns a 1 if the opening and closing parentheses in a character vector are balanced, or a zero otherwise.

Test cases:

```
{your_solution} '((2×3)+4)'
```

1

```
{your_solution} ''
```

1

```
{your_solution} 'hello world!'
```

1

```
{your_solution} ')(2×3)+4('
```

0

```
{your_solution} '(()'
```

0

```
{your_solution} ')'
```

0

Problem 5 - Identity crisis

An identity matrix is a square matrix (table) of 0 with 1's in the main diagonal.

Write an APL dfn which produces an $n \times n$ identity matrix.

Test cases:

```
{your_solution} 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
```

`{your_solution} 1 A` should return a 1×1 matrix
1

`{your_solution} 0 A` should return a 0×0 matrix

Problem 6 - Home on the range

Write a defn which returns the magnitude of the range (i.e. the difference between the lowest and highest values) of a numeric array.

Test cases:

`{your_solution} 19 -3 7.6 22`
25

`{your_solution} 101 A` should work with a scalar argument
0

`{your_solution} 2 3p10 20 30 40 50 60 A` should work with
arrays of any number of dimensions
50

`{your_solution} 10 A` including empty arrays
0

Problem 7 - Float your boat

Write a defn which selects the floating point (non-integer) numbers from a numeric vector.

Test cases:

`{your_solution} 14.2 9 -3 3.1 0 -1.1`
14.2 3.1 -1.1

`{your_solution} 1 3 5 A` should return an empty vector

`{your_solution} 3.1415`
3.1415

Problem 8 - Go forth and multiply

Write a defn which produces a multiplication table.

Test cases:

`{your_solution} 5`
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15

4 8 12 16 20
5 10 15 20 25

{your_solution} 1 A should return a 1x1 matrix
1

{your_solution} 0 A should return a 0x0 matrix

Problem 9 - It's a moving experience

Write a defn which produces n month moving averages for a year's worth of data.

Test cases:

sales ← 200 300 2700 3400 100 2000 400 2100 3500 3000 4700 4300

2 {your_solution} sales A produces 2 month moving averages
250 1500 3050 1750 1050 1200 1250 2800 3250 3850 4500

10 {your_solution} sales A 10 month moving average
1770 2220 2620

1 {your_solution} sales A 1 month moving average is the same
as sales
200 300 2700 3400 100 2000 400 2100 3500 3000 4700 4300

Problem 10 - Solution salvation

Many people have taken some sort of algebra class where you are presented with a set of linear equations like:

$$3x + 2y = 13$$

$$x - y = 1$$

The answer in this case is $x=3$ and $y=2$

Write a defn which solves this type of problem. Hint: this is the easiest of all of the problems presented here.

The left argument is a vector of the values for the equations and the right argument is a matrix of the coefficients.

Test cases:

13 1 {your_solution} 2 2 3 2 1 -1
3 2

2 6 4 {your_solution} 3 3 4 1 3 2 2 2 6 3 1
-1 3 1