Lab session 1: Signals and Systems

Problem 1: Cartesian coordinates to Polar coordinates
Write a program that reads a Cartesian coordinate (x, y) from the input and converts it into the corresponding Polar coordinate (r, θ). The input of this problem consists of two integers: x and y. The output of your program should be two floating point numbers (use `doubles` in your programs): r (length, so a non-negative number) and θ (in radians). The output values should be rounded and have two digits after the decimal dot. Note that, by convention, $-\pi \leq \theta < \pi$.

Example 1:
input:
`2 0`
output:
`2.00 0.00`

Example 2:
input:
`0 2`
output:
`2.00 1.57`

Example 3:
input:
`-2 0`
output:
`2.00 -3.14`

Problem 2: Polar coordinates to Cartesian Coordinates
Write a program that reads a Polar coordinate (r, θ) from the input and converts it into the corresponding Cartesian coordinate (x, y). The input consists of two floating point numbers: r (a non-negative number) and θ (in radians). The output of your program should be two floating point numbers (rounded to two digits after the decimal dot): x and y.

Example 1:
input:
`1.00 1.57`
output:
`0.00 1.00`

Example 2:
input:
`1.00 0.00`
output:
`1.00 0.00`

Example 3:
input:
`1.00 -1.57`
output:
`0.00 -1.00`

Problem 3: Summing two sinusoids having the same frequency
The input of this problem consists of 5 floating point values: f (in Hz), A_0, φ_0 (in radians), A_1, and φ_1 (in radians). Write a program that outputs the equation for the sum of the signals $A_0 \cos(2\pi f t + \varphi_0) + A_1 \cos(2\pi f t + \varphi_1)$ in the form $x(t) = A \cos(2\pi f t + \varphi)$. Note that the equation should be $x(t) = 0.00$ if $A = 0$. All numbers in the output must be rounded and have two digits after the decimal dot.

Example 1:
input:
`42 1 1.047198 1 0.523599`
output:
`x(t)=1.93*cos(2*pi*42.00*t+0.79)`

Example 2:
input:
`1 42 0 42 -3.1415927`
output:
`x(t)=0.00`

Example 3:
input:
`42 100 0.78540 100 2.35619`
output:
`x(t)=141.42*cos(2*pi*42.00*t+1.57)`

Problem 4: Uniform sampling of a sinusoid
The input of this problem consists of 5 values: f_s (sampling rate, integer), n (number of samples, integer), A (amplitude), f (in Hz), and φ (in radians). Write a program that outputs n samples of the discrete time signal that is obtained after uniform sampling of $x(t) = A \cos(2\pi f t + \varphi)$ using the sampling rate f_s. The samples are integers. Do not round the samples, but rely on C's conversion of floating point values to integers (i.e. truncation). Make sure that your output has the same format as the folllowing examples.

Example 1:
input:
`10 10 42 10 0`
output:
`[42,42,42,42,42,42,42,42,42,42]`

Example 2:
input:
`20 10 42 10 0`
output:
`[42,-42,42,-42,42,-42,42,-42,42,-42]`

Example 3:
input:
`4 10 1 1 -3.14159265`
output:
`[-1,0,1,0,-1,0,1,0,-1,0]`

Problem 5: Uniform sampling of a sum of sinusoids

The input of this problem consists of several lines. The first line contains two integers: f_s (sampling rate), and n (number of samples). The second line contains a positive integer m (number of components). Next follow m lines, each containing three values describing a sinusoid: A (amplitude), f (in Hz), and φ (in radians). Write a program that outputs n samples of the discrete time signal that is obtained after uniform sampling the sum of the m sinusoids. As in the previous exercise, samples are integers and should not be rounded.

Example 1:

input:
```
20 10
1
42 10 0
```
output:
```
[42,-42,42,-42,42,-42,42,-42,42,-42]
```

Example 2:

input:
```
20 10
2
42 10 0
42 10 0
```
output:
```
[84,-84,84,-84,84,-84,84,-84,84,-84]
```

Example 3:

input:
```
20 10
2
42 10 0
42 10 -3.14159265
```
output:
```
[0,0,0,0,0,0,0,0,0,0]
```

Problem 6: Aliases?

The input of this problem consists of three lines. The first line contains a positive integer f_s (the sampling rate). The remaining two lines each contain the parameters that define a sinusoid: A (amplitude), f (in Hz), and φ (in radians). Write a program that outputs `YES` if uniform sampling of the two sinusoids using the given sampling rate yields the same discrete output. Otherwise, it should output `NO`.

Example 1:

input:
```
100
1 10 0
1 20 0
```
output:
```
NO
```

Example 2:

input:
```
100
1 100 0
1 200 0
```
output:
```
YES
```

Example 3:

input:
```
100
50 120 0
50 80 0
```
output:
```
YES
```

Problem 7: Microphones

Two sound sources are placed on the x-axis, i.e. at locations $(x_0, 0)$, and $(x_1, 0)$. Both sources produce the same sinusoid. Next, a microphone is also placed on the x-axis at location $(x_2, 0)$. The objective is to compute the amplitude of the sinusoid that is observed by the microphone. Take the speed of sound to be 343 m/s, and ignore sound degradation.

The input of this problem consist of two lines. The first line defines the sinusoid: A (amplitude), f (in Hz), and φ (in radians). The second lines contains the three locations x_0, x_1 and x_2 (in meters). The output of the program must be the amplitude of the observed signal, rounded to two digits after the decimal dot.

Example 1:

input:
```
5 100 0
-100 100 0
```
output:
```
10.00
```

Example 2:

input:
```
5 100 0
-10 10 1
```
output:
```
2.58
```

Example 3:

input:
```
5 100 0
-10 10 -1
```
output:
```
2.58
```

Problem 8: Periodic signal

The input of this problem consists of several lines. The first line contains a positive integer n. Next follow n lines, each containing the parameters that define a sinusoid: A (amplitude), T (period, integer, in microseconds), and φ (in radians). The sum of the n sinusoids is also periodic. The output of your program must be the fundamental period T_0 (integer, in microseconds) of this sum.

Example 1:

input:
```
4
4 3 0.79
9 6 1.38
8 9 2.62
6 4 0.38
```
output:
```
36
```

Example 2:

input:
```
10
5 6 2.51
1 2 1.19
9 6 2.09
1 9 1.40
4 8 0.42
3 5 1.19
8 4 1.13
8 3 0.80
8 6 0.51
6 10 0.26
```
output:
```
360
```

Example 3:

input:
```
10
4 1 1.12
8 4 1.64
5 10 2.06
1 7 0.35
5 4 2.69
4 5 0.84
3 4 1.76
3 7 0.63
6 7 2.62
8 1 2.25
```
output:
```
140
```

Problem 9: Sum to Product

The input of this problem consists of two frequencies: f_0 and f_1 (both in Hz, integer). These input values define the signal $x(t) = \cos(2\pi f_0 t) + cos(2\pi f_1 t)$. This signal is a beat note (assuming $f_0 \neq f_1$), which can be rewritten as $x(t) = A\cos(2\pi f_2 t)\cos(2\pi f_3 t)$. The output of your program must be the frequencies f_2 and f_3 (where $f_2 < f_3$). The output must be rounded to two digits after the decimal dot.

Example 1:

input:
```
672 648
```
output:
```
12.00 660.00
```

Example 2:

input:
```
362 134
```
output:
```
114.00 248.00
```

Example 3:

input:
```
919 732
```
output:
```
93.50 825.50
```