# Cosine of Difference

Given a vector $\mathbf{x} \in \mathbb{R}^n$ where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, create the matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ whose elements are $a_{ij} = \cos(x_i - x_j)$.

The resulting matrix $\mathbf{A}$ has the form:

$$\mathbf{A} = \begin{pmatrix} \cos(x_1 - x_1) & \cos(x_1 - x_2) & \cdots & \cos(x_1 - x_n) \\ \cos(x_2 - x_1) & \cos(x_2 - x_2) & \cdots & \cos(x_2 - x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(x_n - x_1) & \cos(x_n - x_2) & \cdots & \cos(x_n - x_n) \end{pmatrix}$$

## Input Format

1d numpy array of size $n$ containing real numbers $x_1, x_2, \ldots, x_n$

## Output Format

Output the resulting $n \times n$ matrix $\mathbf{A}$ where each element $a_{ij} = \cos(x_i - x_j)$. Each element should be returned as a floating-point number.

## Constraints

- $\mathbf{x} \in \mathbb{R}^n$
- $1 \leq n \leq 1000$
- All input values are real numbers in the range $[-100, 100]$

## Sample Input

```
x = [0, np.pi/2, np.pi] # np.pi is a pi constant in numpy
```

## Sample Output

```
[[1.0, 0.0, -1.0],
 [0.0, 1.0, 0.0],
 [-1.0, 0.0, 1.0]]
```

## Implementation

**Goal:** Fill in the following function:

```
def cosine_of_difference(x):
    ...
    return ... # Return the resulting matrix
exec("\n".join(iter(input, "#Exit"))) # Don’t remove this line
```

## Hint

- `np.cos` to compute the cosine of a number (element-wise)
- numpy broadcasting (`https://numpy.org/doc/stable/user/basics.broadcasting.html`)