

HW2

January 25, 2026

1 Homework 2

1.0.1 Wing Huang PID: A18894844

Date: 2026-01-20

1.0.2 Problem 1 : More Challenging Functions

(a)

```
[18]: def mysum(N):  
        if N==1:  
            return 1  
        else:  
            return N+mysum(N-1)
```

```
[19]: print(mysum(10))  
        print(mysum(5))  
        print(mysum(1))
```

55

15

1

(b)

```
[20]: def sort(items):  
        nums = [x for x in items if isinstance(x, (int, float))]  
        strs = [x for x in items if isinstance(x, str)]  
        return sorted(nums) + sorted(strs)
```

```
[21]: test = [78, 'question', 6.9, 'mac', 114514, 'nvida', -1]  
        sort(test)
```

```
[21]: [-1, 6.9, 78, 114514, 'mac', 'nvida', 'question']
```

(c)

```
[22]: def calc(lst):  
        result = lst[0]
```

```

for i in range(1, len(lst), 2):
    if lst[i] == '+':
        result += lst[i+1]
    elif lst[i] == '-':
        result -= lst[i+1]
    elif lst[i] == '*':
        result *= lst[i+1]
    elif lst[i] == '/':
        result /= lst[i+1]
return result

```

```

[23]: list1c = [2, '+', 4, '-', 5, '*', 10]
      calc(list1c)

```

[23]: 10

(d)

```

[24]: def calc2(lst):
      t = [lst[0]]
      i = 1
      while i < len(lst):
          op = lst[i]
          num = lst[i+1]
          if op in ('*', '/'):
              left = t.pop()
              t.append(left * num if op == '*' else left / num)
          else:
              t.append(op)
              t.append(num)
          i += 2

      value = t[0]
      i = 1
      while i < len(t):
          op = t[i]
          num = t[i+1]
          if op == '+':
              value += num
          elif op == '-':
              value -= num
          i += 2

      return value

```

```

[25]: list1d = [2, '+', 4, '-', 5, '*', 10]
      calc2(list1d)

```

[25]: -44

1.0.3 Problem 2 : Basic numpy

(a)

```
[26]: import numpy as np

def sinn(t, n, omega, phi):
    return np.sin((omega*t)**n+phi)
```

```
[27]: time = np.linspace(0, 10, 10)
sinn(time, 1, 10, 1)
```

```
[27]: array([ 0.84147098, -0.4396954 , -0.94284711,  0.22231267,  0.9941035 ,
            0.00688774, -0.99251546, -0.235722 ,  0.93816742,  0.45202579])
```

(b)

```
[28]: def solve_quad(a, b, c):
    a = np.asarray(a)
    b = np.asarray(b)
    c = np.asarray(c)

    disc = b*b - 4*a*c
    disc = disc.astype(np.complex128)
    sqrt_disc = np.sqrt(disc)

    x1 = (-b + sqrt_disc) / (2*a)
    x2 = (-b - sqrt_disc) / (2*a)
    return x1, x2
```

```
[29]: print(solve_quad(1, 0, 1), "\n")
print(np.array([11,1,1]), np.array([6,7,2]), np.array([1,1,1]))
```

```
(np.complex128(1j), np.complex128(-1j))
```

```
[11  1  1] [6 7 2] [1 1 1]
```

(c)

```
[30]: x = np.linspace(-np.pi, np.pi, 10)
y = np.linspace(0, 2*np.pi, 10)
z = np.linspace(-3, 3, 10)

X, Y, Z = np.meshgrid(x, y, z, indexing="ij")

F = np.sin(X) * (np.cos(Y)**2) + Z
```

```
F.shape
print(F[1,1,1])

i, j, k = 3, 4, 5
print(x[i], y[j], z[k])
print(F[i, j, k])
```

```
-2.710536586701078
-1.0471975511965979 2.792526803190927 0.33333333333333304
-0.4313863426433554
```

```
[ ]:
```