class polynomial:

def \_\_init\_\_(self,coeffs):

self.coeffs=coeffs

def evaluate(self,x):

result=0

for exp,coeff in self.coeffs.items():

result+=coeff\*(x\*\*exp)

return result

def \_\_add\_\_(self,other):

result\_coeffs={}

for exp,coeff in self.coeffs.items():

result\_coeffs[exp]=coeff

for exp,coeff in other.coeffs.items():

result\_coeffs[exp]=result\_coeffs.get(exp,0)+coeff

return polynomial(result\_coeffs)

def \_\_str\_\_(self):

terms=[]

for exp,coeff in self.coeffs.items():

if exp==0:

term=str(coeff)

elif exp==1:

term=f"{coeff}x"

else:

term=f"{coeff}x^{exp}"

terms.append(term)

return"+".join(terms)

poly1=polynomial({2:3,1:2,0:5})

poly2=polynomial({2:2,1:-1,0:3})

print("polynomial1:",poly1)

print("polynomial2:",poly2)

sum\_poly=poly1 + poly2

print("sum:",sum\_poly)

x\_value=2

print(f"evaluating at x={x\_value}:")

print("poly 1:",poly1.evaluate(x\_value))

print("poly 2:",poly2.evaluate(x\_value))

print("sum:",sum\_poly.evaluate(x\_value))