Homework #2

Wenyi Liu

Question a & b & c

```
/* Node for the singly linked list */
class Node {

public:
  double data;
  Node *next;

  Node() {
    data = 0.0;
    next = nullptr;
  }

  Node(double inData) : data(inData), next(nullptr) {}
};


/* Singly linked list */
class List {

public:
  int size;
  Node *head;
  Node *tail;

  List() {
```

```
  size = 0;
  head = nullptr;
  tail = nullptr;
}

// destructor, free all nodes on the heap
~List() {
  Node *cur = head;
  while (cur) {
    Node *next = cur->next;
    delete cur;
    cur = next;
  }
  head = nullptr;
  tail = nullptr;
}

/* add a new node with value to the end of the list */
void add(double value) {
  if (size == 0) {
    head = new Node(value);
    tail = head;
  } else {
    tail->next = new Node(value);
    tail = tail->next;
  }
  ++size;
```

```
  }
};


/* append a constant to the polynomial */
void appendTerm(List *pPolynomial, double constant) {
  pPolynomial->add(constant);
}


/* disply the polynomial to stdout */
void display(List *pPolynomial) {
  Node *head = pPolynomial->head;
  int power = pPolynomial->size - 1; // this is the exponent

  // traverse the list from head to tail
  while (head) {
    double term = head->data;
    if (term > 0) {
      // need a plus sign if not the first term
      if (head != pPolynomial->head) {
        cout << "+ ";
      }
      // determine if we need x^n, x, or just the term
      if (power > 1) {
        if (term != 1) { cout << term; }
        cout << "x^" << power << " ";
      } else if (power == 1) {
```

```
      if (term != 1) { cout << term; }
      cout << "x ";
    } else {
      cout << term << " ";
    }
  } else if (term < 0) {
    // need a minus sign if not the first term
    if (head != pPolynomial->head) {
      cout << "- ";
    } else {
      cout << "-";
    }
    // determine if we need x^n, x, or just the term
    if (power > 1) {
      if (term != -1) { cout << -term; }
      cout << "x^" << power << " ";
    } else if (power == 1) {
      if (term != -1) { cout << -term; }
      cout << "x ";
    } else {
      cout << -term << " ";
    }
  }
  // if the term is 0, nothing to output
  head = head->next;
  --power;
}
```

```
}


/* evaluate the polynomial given the input x */
double evaluate(List *pPolynomial, double x) {
  Node *head = pPolynomial->head;
  int power = pPolynomial->size - 1;
  double result = 0.0;
  // traverse the list from head to tail
  // evaluate each term and add them up
  while (head) {
    result += head->data * pow(x, power);
    head = head->next;
    --power;
  }
  return result;
}
```

Question d

```
/* main function */
int main() {


  // test 1: x + 1.0
  vector<double> coefficient1 { 1.0, 1.0 };


  // create the polynomial
  List *polynomial1 = new List();
  for (double constant : coefficient1) {
```

```
  appendTerm(polynomial1, constant);
}
// display and evaluate the polynomial
cout << "==============Test case 1==============\n";
cout << "Evaluate ";
display(polynomial1);
cout << "with x = 1.0\n";
cout << "Answer is " << evaluate(polynomial1, 1.0) << "\n";

delete polynomial1;

// test 2: x^2 - 1.0
vector<double> coefficient2 { 1.0, 0.0, -1.0 };

// create the polynomial
List *polynomial2 = new List();
for (double constant : coefficient2) {
  appendTerm(polynomial2, constant);
}
// display and evaluate the polynomial
cout << "==============Test case 2==============\n";
cout << "Evaluate ";
display(polynomial2);
cout << "with x = 2.03\n";
cout << "Answer is " << evaluate(polynomial2, 2.03) << "\n";

delete polynomial2;
```

```
// test 3: -3.0x^3 + 0.5x^2 - 2.0x
vector<double> coefficient3 { -3.0, 0.5, -2.0, 0.0 };

// create the polynomial
List *polynomial3 = new List();
for (double constant : coefficient3) {
  appendTerm(polynomial3, constant);
}
// display and evaluate the polynomial
cout << "==============Test case 3==============\n";
cout << "Evaluate ";
display(polynomial3);
cout << "with x = 5.0\n";
cout << "Answer is " << evaluate(polynomial3, 5.0) << "\n";

delete polynomial3;

// test 4: -0.3125x^4 - 9.915x^2 - 7.75x - 40.0
vector<double> coefficient4 { -0.3125, 0.0, -9.915, -7.75, -40.0 };

// create the polynomial
List *polynomial4 = new List();
for (double constant : coefficient4) {
  appendTerm(polynomial4, constant);
}
// display and evaluate the polynomial
```

```
  cout << "==============Test case 4==============\n";
  cout << "Evaluate ";
  display(polynomial4);
  cout << "with x = 123.45\n";
  cout << "Answer is " << evaluate(polynomial4, 123.45) << "\n";

  delete polynomial4;

  return 0;
}
```

Output:

```
==============Test case 1==============
Evaluate x + 1 with x = 1.0
Answer is 2
==============Test case 2==============
Evaluate x^2 - 1 with x = 2.03
Answer is 3.1209
==============Test case 3==============
Evaluate -3x^3 + 0.5x^2 - 2x with x = 5.0
Answer is -372.5
==============Test case 4==============
Evaluate -0.3125x^4 - 9.915x^2 - 7.75x - 40 with x = 123.45
Answer is -7.27317e+07
```