202001492 김수민 프로그래밍 입문 12번 과제

1. 메뉴 설계 다음과 같이 다항식 연산을 수행할 수 있는 메뉴를 작성한다.

-----------------------------------------------------------------------------

1. 다항식 입력

2. 다항식 출력

3. 다항식 계산

4. 다항식 덧셈

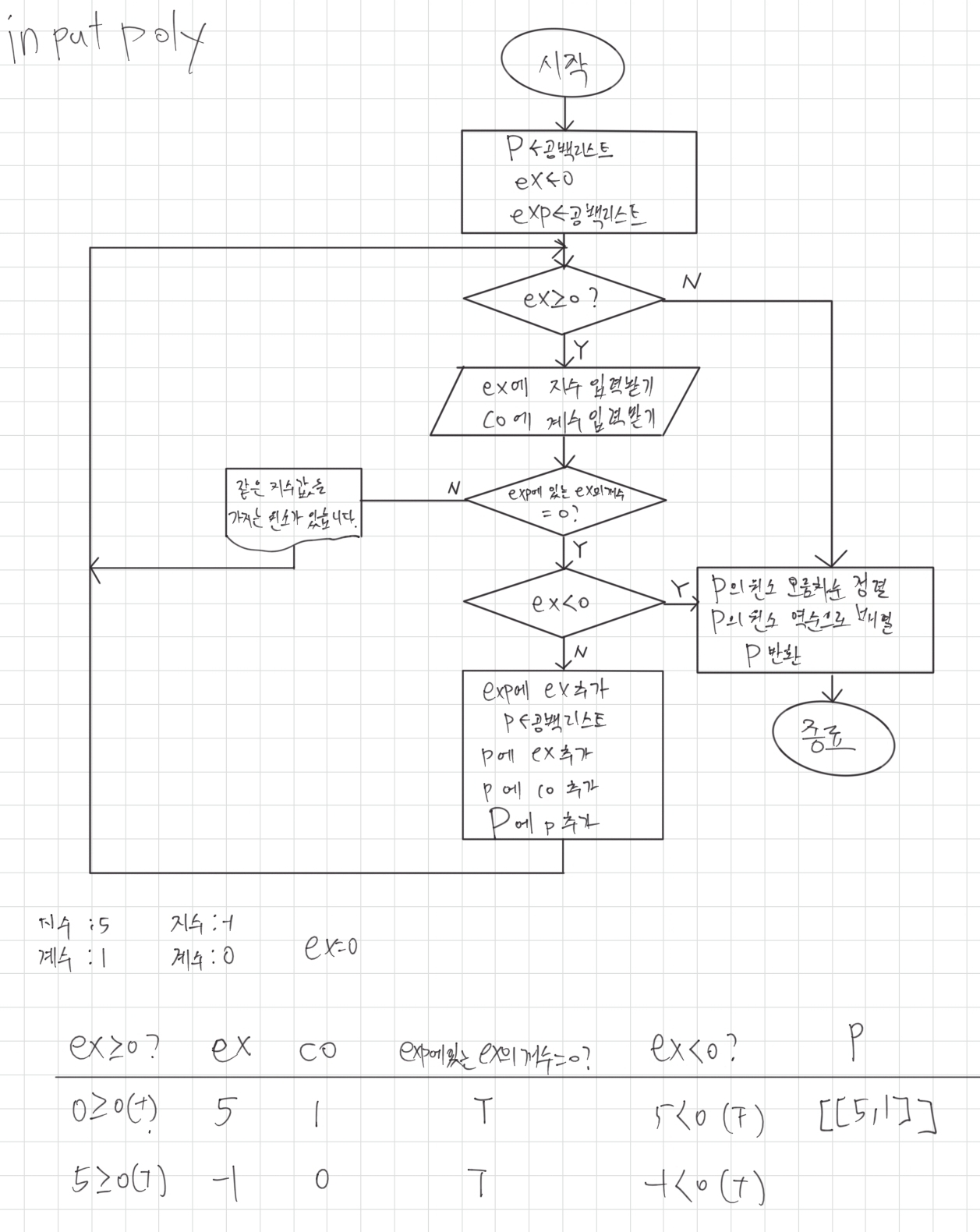
5. 다항식 곱셈 메뉴 선택 (종료시는 9) :

-----------------------------------------------------------------------------

이 프로그램은 poly.py에 저장되어 있다.

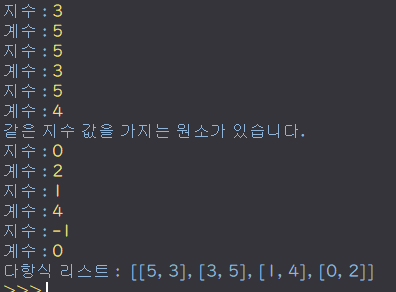
2. 다항식을 입력하는 함수 inputPoly()를 작성하라. 다항식을 표현하는 리스트는 지 수에 대한 내림차순으로 정렬되어 있어야 하며, 지수는 하나의 다항식 내에서 유 일한 값을 가져야 한다. 따라서 동일한 값을 갖는 지수를 입력하려고 하면 “같 은 지수 값을 가지는 원소가 있습니다.”라는 오류 메시지를 출력한다. 이 함수의 입력과 출력 예는 다음과 같다. 지수는 음이 아닌 정수 값을 가져야 하므로 지수에 음수 값을 입력하면 입력 함수를 종료한다.

<순서도와 표>



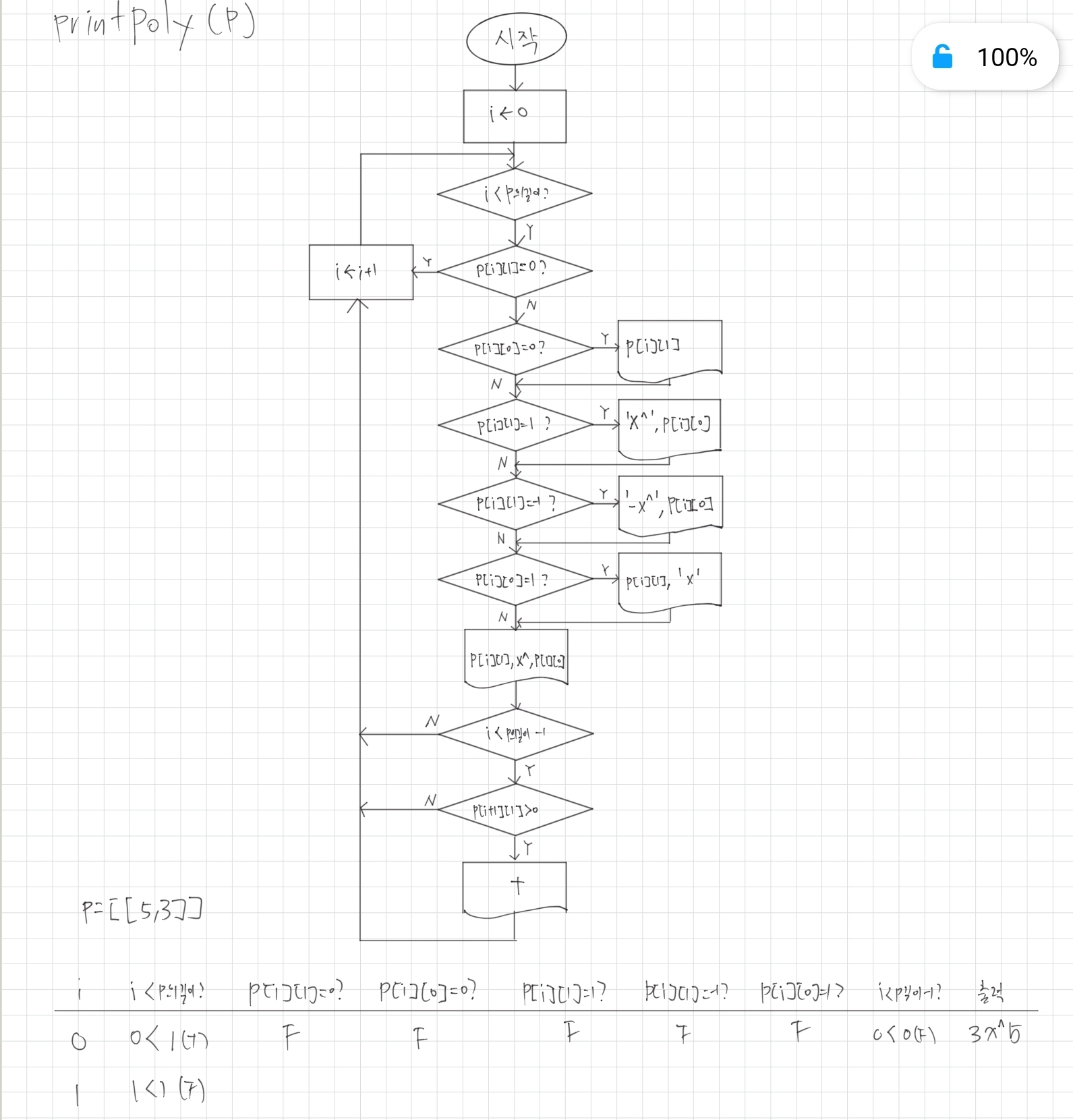
<코드와 실행결과>

|  |
| --- |
| def inputPoly():  P = []  ex = 0  exp = []  while ex >= 0:  ex = int(input('지수 : '))  co = int(input('계수 : '))  if exp.count(ex) == 0:  if ex < 0 : break  exp.append(ex)  p = []  p.append(ex)  p.append(co)  P.append(p)  else : print('같은 지수 값을 가지는 원소가 있습니다.')  P.sort()  P.reverse()  return P  print('다항식 리스트 : ', inputPoly()) |



3. 다항식을 표현하는 리스트 p를 다항식 형태로 출력하는 함수 printPoly(p)를 작성 하라. 입력 받은 다항식 리스트가 [[5, 3], [3, 5], [1, 4], [0, 2]]이라고 할 때, 출력 되는 다항식은 다음과 같다.

<순서도와 표>



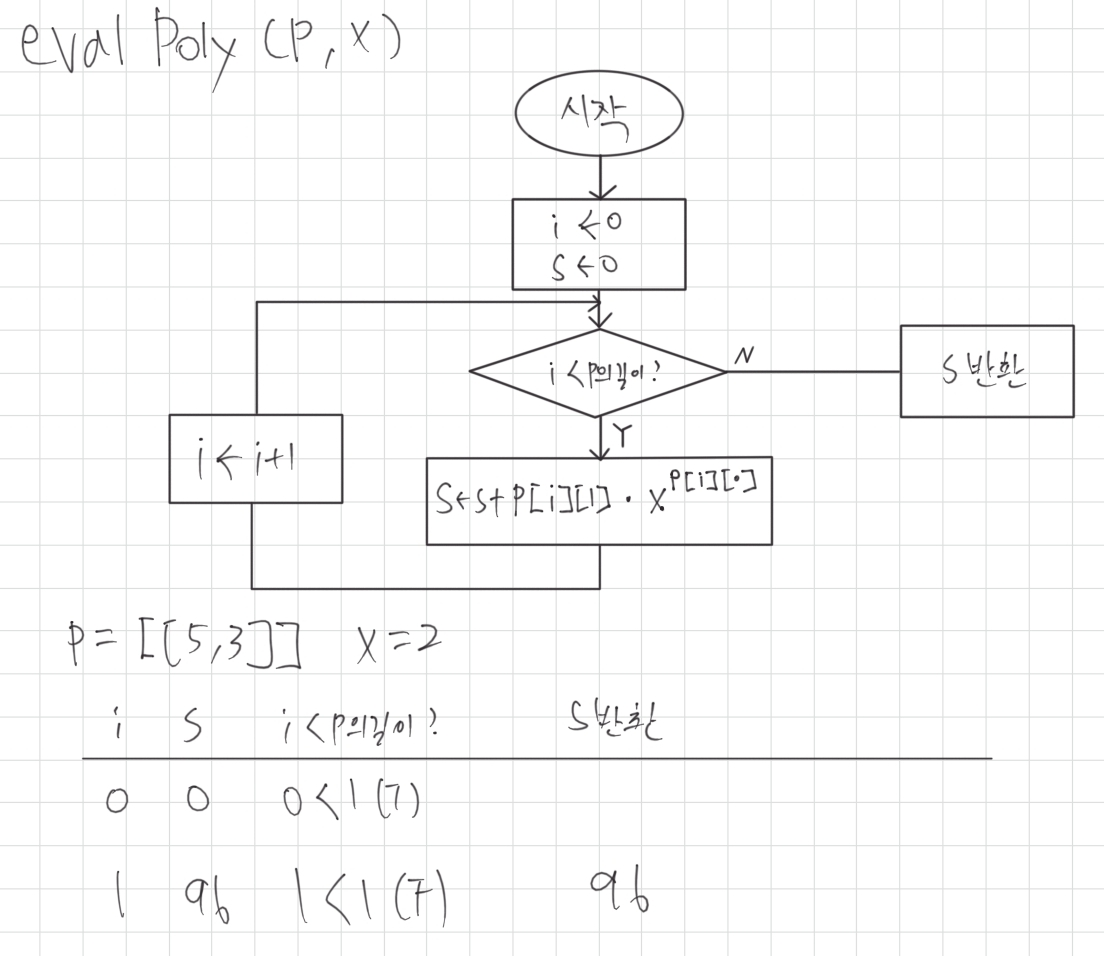
<코드와 실행결과>

|  |
| --- |
| def printPoly(p):  for i in range (len(p)):  if p[i][1] == 0: continue  if p[i][0] == 0:  print(p[i][1], end = '')  elif p[i][1] == 1: print('x^{}'.format(p[i][0]), end=' ')  elif p[i][1] == -1 : print('-x^{}'.format(p[i][0]), end=' ')  elif p[i][0] == 1: print('{}x'.format(p[i][1]), end=' ')  else : print('{}x^{}'.format(p[i][1], p[i][0]), end=' ')  if i < len(p)-1:  if p[i+1][1] > 0 :  print('+', end = ' ')  p = [[5, 3], [3, 5], [1, 4], [0, 2]]  print('다항식 리스트 : ', p)  print('다항식 = ',end='')  printPoly(p) |



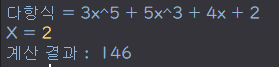
4. x의 값을 입력하면 다항식을 계산한 결과를 반환하는 함수 evalPoly(p)를 작성하 라. 다음은 다항식 계산의 예를 보여주고 있다.

<순서도와 표>



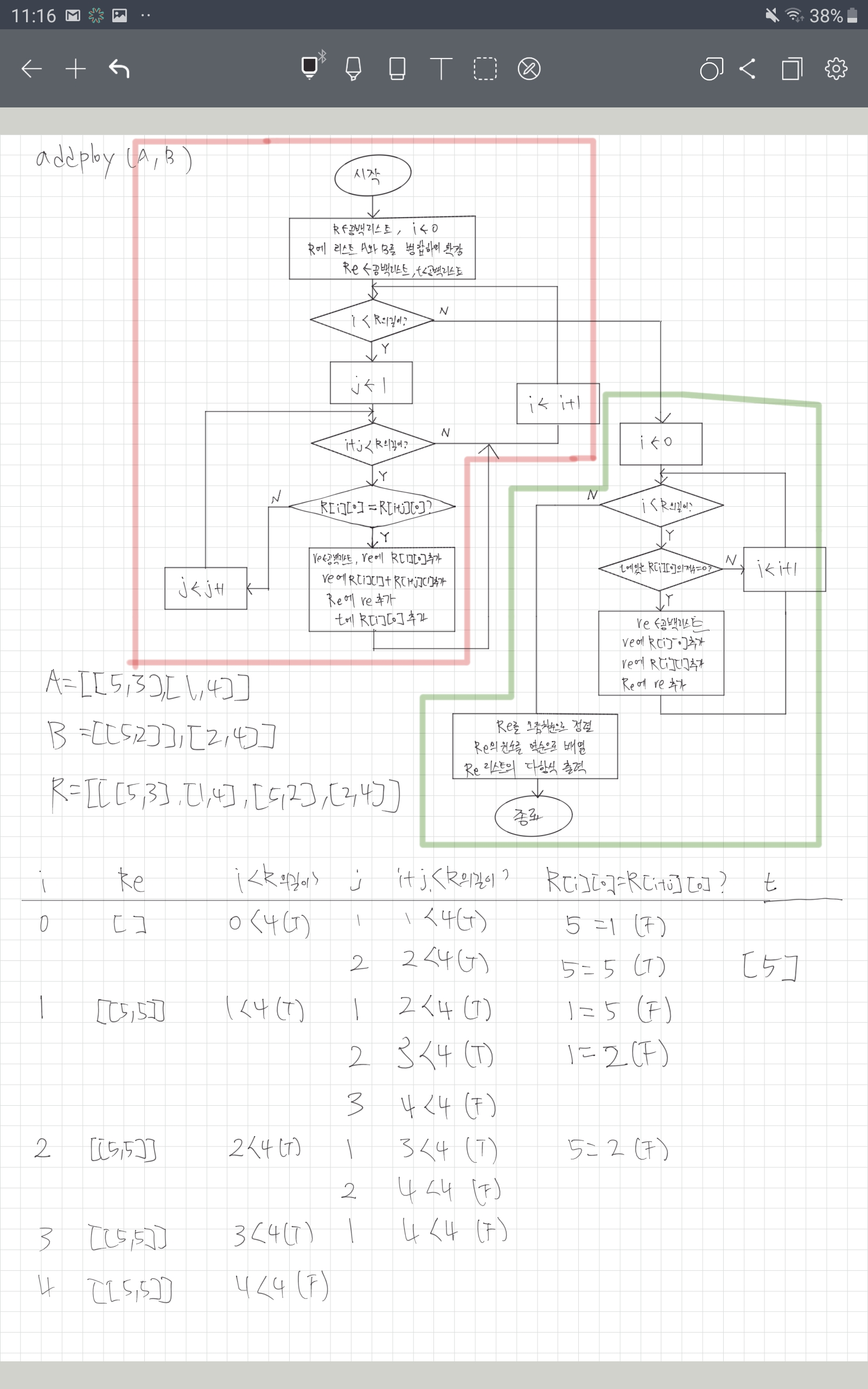
<코드와 실행결과>

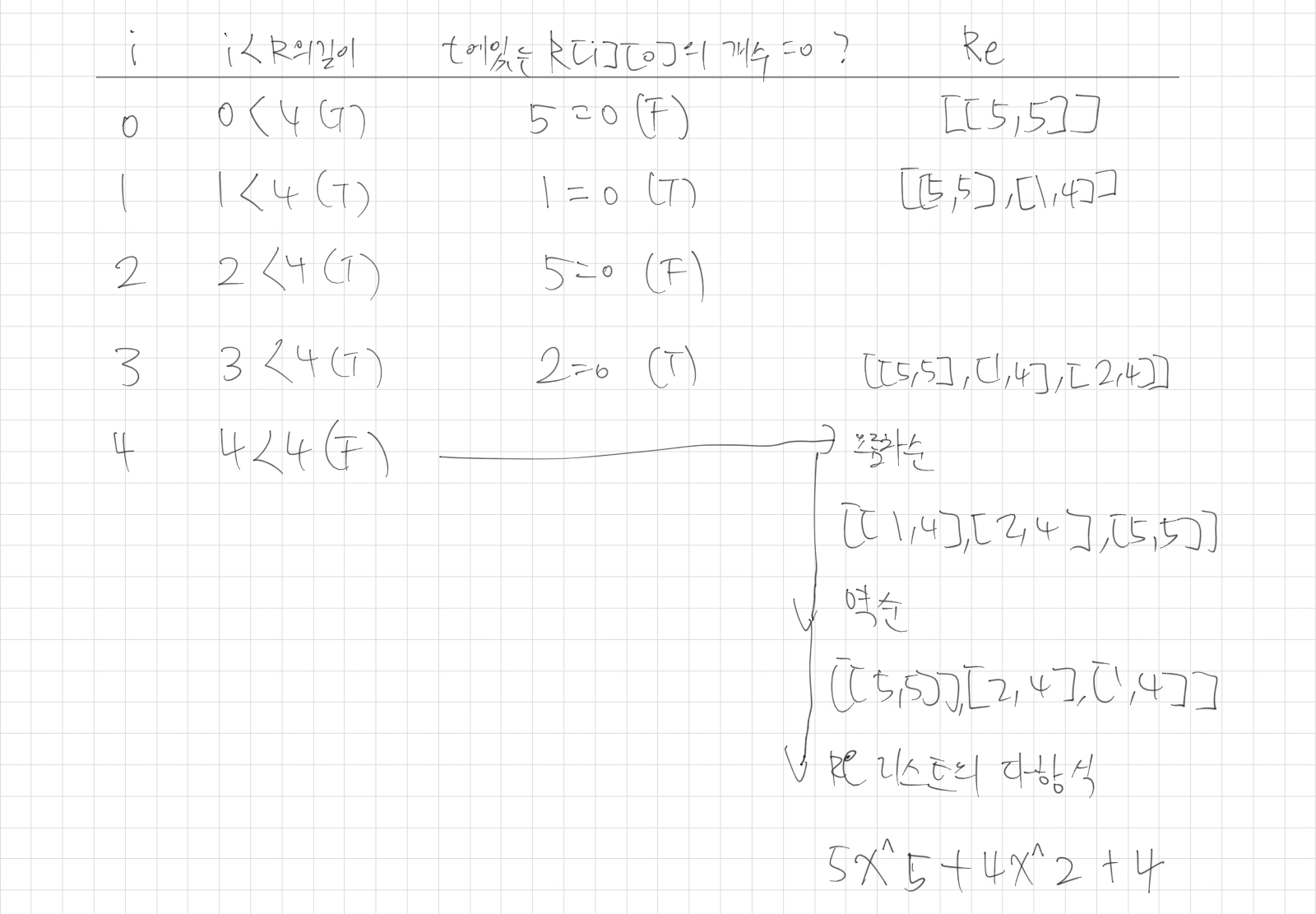
|  |
| --- |
| def printPoly(p):  for i in range (len(p)):  if p[i][1] == 0: continue  if p[i][0] == 0:  print(p[i][1], end = '')  elif p[i][1] == 1: print('x^{}'.format(p[i][0]), end=' ')  elif p[i][1] == -1 : print('-x^{}'.format(p[i][0]), end=' ')  elif p[i][0] == 1: print('{}x'.format(p[i][1]), end=' ')  else : print('{}x^{}'.format(p[i][1], p[i][0]), end=' ')  if i < len(p)-1:  if p[i+1][1] > 0 :  print('+', end = ' ')    def evalPoly(p, x):  s = 0  for i in range(len(p)):  s += p[i][1]\*x\*\*p[i][0]  return s  p = [[5, 3], [3, 5], [1, 4], [0, 2]]  print('다항식 = ', end='')  printPoly(p)  print()  x = int(input('X = '))  print('계산 결과 : ', evalPoly(p, x)) |



5. 두 개의 다항식 A와 B를 더한 결과를 반환하는 함수 addPoly(A, B)를 작성하라. 다항식을 더할 때에는 지수가 같은 항끼리 계수를 서로 더한다. 다음은 다항식 덧셈의 예를 보여주고 있다.

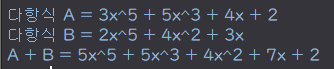
<순서도와 표>





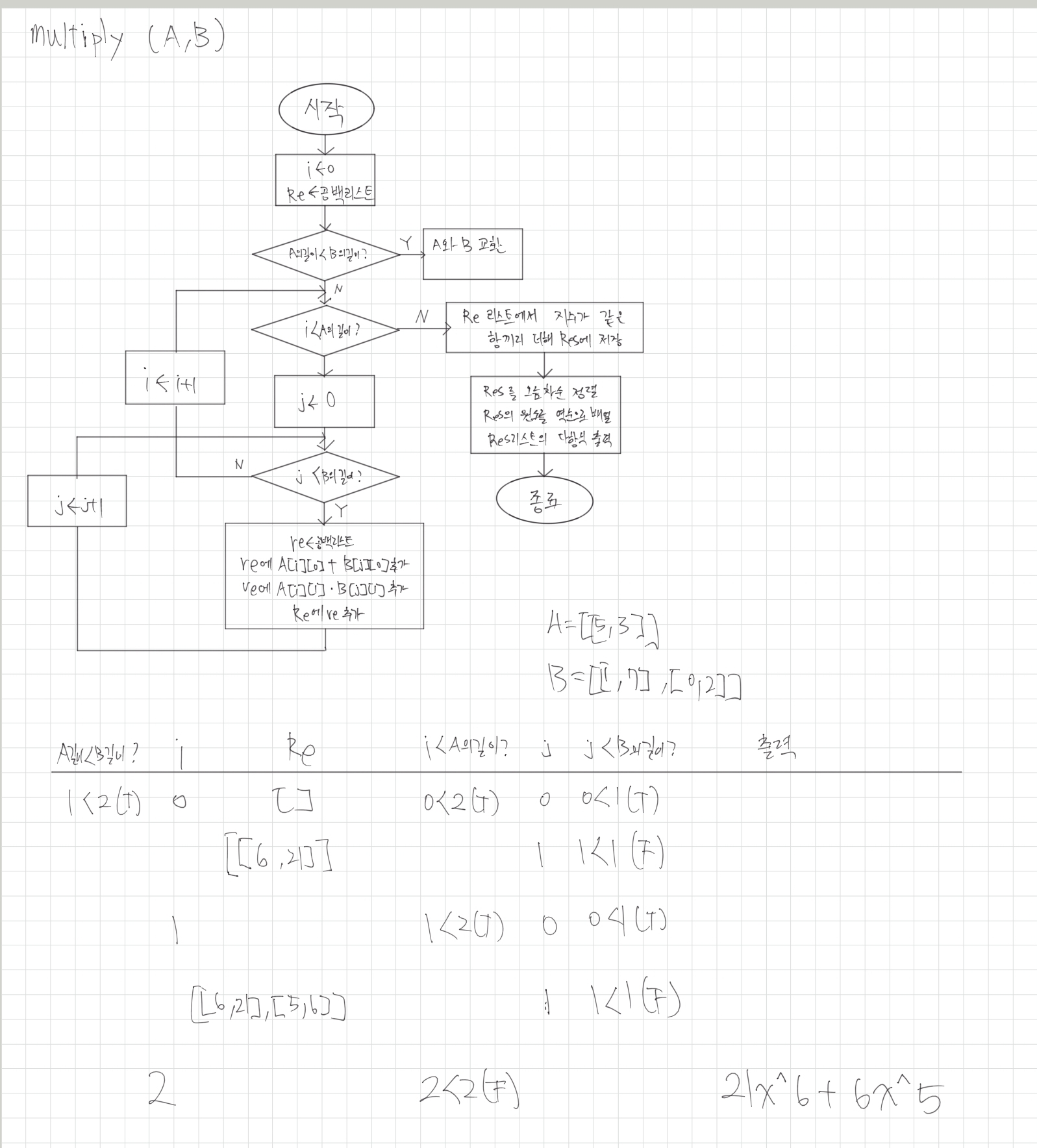
<코드와 실행결과>

|  |
| --- |
| def printPoly(p):  for i in range (len(p)):  if p[i][1] == 0: continue  if p[i][0] == 0:  print(p[i][1], end = '')  elif p[i][1] == 1: print('x^{}'.format(p[i][0]), end=' ')  elif p[i][1] == -1 : print('-x^{}'.format(p[i][0]), end=' ')  elif p[i][0] == 1: print('{}x'.format(p[i][1]), end=' ')  else : print('{}x^{}'.format(p[i][1], p[i][0]), end=' ')  if i < len(p)-1:  if p[i+1][1] > 0 :  print('+', end = ' ')    def addPoly(A, B):  R = []  R.extend(A)  R.extend(B)  Re = []  t = []  for i in range(len(R)):  j = 1  while i+j < len(R):  if R[i][0] == R[i+j][0]:  re = []  re.append(R[i][0])  re.append(R[i][1]+R[i+j][1])  Re.append(re)  t.append(R[i][0])  break  j += 1  for i in range(len(R)):  if t.count(R[i][0]) == 0:  re=[]  re.append(R[i][0])  re.append(R[i][1])  Re.append(re)  Re.sort()  Re.reverse()  printPoly(Re)  A = [[5, 3], [3, 5], [1, 4], [0, 2]]  B = [[5, 2], [2, 4], [1, 3]]  print('다항식 A = ',end='')  printPoly(A)  print()  print('다항식 B = ',end='')  printPoly(B)  print()  print('A + B = ',end='')  addPoly(A,B) |



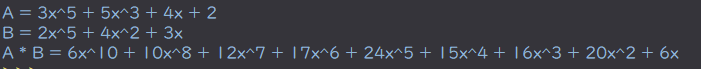
6. 두 개의 다항식 A와 B를 곱한 결과를 반환하는 함수 multiplyPoly(A, B)를 작성하 라. 다항식을 곱할 때에는 A의 모든 항과 B의 모든 항을 차례대로 곱하며, 계수 는 곱하고 지수는 더해서 새로운 결과 다항식을 만든다. 결과 다항식에서 지수가 같은 항이 있는 경우는 계수를 서로 더해서 간단하게 만든다. 다음은 다항식 곱 셈의 예를 보여주고 있다.

<순서도와 표>



<코드와 실행결과>

|  |
| --- |
| def printPoly(p):  for i in range (len(p)):  if p[i][1] == 0: continue  if p[i][0] == 0:  print(p[i][1], end = '')  elif p[i][1] == 1: print('x^{}'.format(p[i][0]), end=' ')  elif p[i][1] == -1 : print('-x^{}'.format(p[i][0]), end=' ')  elif p[i][0] == 1: print('{}x'.format(p[i][1]), end=' ')  else : print('{}x^{}'.format(p[i][1], p[i][0]), end=' ')  if i < len(p)-1:  if p[i+1][1] > 0 :  print('+', end = ' ')    def multiplyPoly(A, B):  Re = []  if len(A)<len(B):  A, B = B, A  for i in range(len(A)):  for j in range(len(B)):  re = []  re.append(A[i][0]+B[j][0])  re.append(A[i][1]\*B[j][1])  Re.append(re)  t = []  Res=[]  for i in range(len(Re)):  j = 1  while i+j < len(Re):  if Re[i][0] == Re[i+j][0]:  res = []  res.append(Re[i][0])  res.append(Re[i][1]+Re[i+j][1])  Res.append(res)  t.append(Re[i][0])  break  j += 1  for i in range(len(Re)):  if t.count(Re[i][0]) == 0:  res=[]  res.append(Re[i][0])  res.append(Re[i][1])  Res.append(res)  Res.sort()  Res.reverse()  printPoly(Res)      A = [[5, 3], [3, 5], [1, 4], [0, 2]]  B = [[5, 2], [2, 4], [1, 3]]  print('A = ',end='')  printPoly(A)  print()  print('B = ',end='')  printPoly(B)  print()  print('A \* B = ',end='')  multiplyPoly(A, B) |



7. 전체 프로그램

<코드와 실행결과>

|  |
| --- |
| def inputPoly():  P = []  ex = 0  exp = []  while ex >= 0:  ex = int(input('지수 : '))  co = int(input('계수 : '))  if exp.count(ex) == 0:  if ex < 0 : break  exp.append(ex)  p = []  p.append(ex)  p.append(co)  P.append(p)  else : print('같은 지수 값을 가지는 원소가 있습니다.')  P.sort()  P.reverse()  return P    def printPoly(p):  for i in range (len(p)):  if p[i][1] == 0: continue  if p[i][0] == 0:  print(p[i][1], end = '')  elif p[i][1] == 1: print('x^{}'.format(p[i][0]), end=' ')  elif p[i][1] == -1 : print('-x^{}'.format(p[i][0]), end=' ')  elif p[i][0] == 1: print('{}x'.format(p[i][1]), end=' ')  else : print('{}x^{}'.format(p[i][1], p[i][0]), end=' ')  if i < len(p)-1:  if p[i+1][1] > 0 :  print('+', end = ' ')    def evalPoly(p, x):  s = 0  for i in range(len(p)):  s += p[i][1]\*x\*\*p[i][0]  return s  def addPoly(A, B):  R = []  R.extend(A)  R.extend(B)  Re = []  t = []  for i in range(len(R)):  j = 1  while i+j < len(R):  if R[i][0] == R[i+j][0]:  re = []  re.append(R[i][0])  re.append(R[i][1]+R[i+j][1])  Re.append(re)  t.append(R[i][0])  break  j += 1  for i in range(len(R)):  if t.count(R[i][0]) == 0:  re=[]  re.append(R[i][0])  re.append(R[i][1])  Re.append(re)  Re.sort()  Re.reverse()  printPoly(Re)      def multiplyPoly(A, B):  Re = []  if len(A)<len(B):  A, B = B, A  for i in range(len(A)):  for j in range(len(B)):  re = []  re.append(A[i][0]+B[j][0])  re.append(A[i][1]\*B[j][1])  Re.append(re)  t = []  Res=[]  for i in range(len(Re)):  j = 1  while i+j < len(Re):  if Re[i][0] == Re[i+j][0]:  res = []  res.append(Re[i][0])  res.append(Re[i][1]+Re[i+j][1])  Res.append(res)  t.append(Re[i][0])  break  j += 1  for i in range(len(Re)):  if t.count(Re[i][0]) == 0:  res=[]  res.append(Re[i][0])  res.append(Re[i][1])  Res.append(res)  Res.sort()  Res.reverse()  printPoly(Res)    m = 1  while m != 9:  print('1. 다항식 입력')  print('2. 다항식 출력')  print('3. 다항식 계산')  print('4. 다항식 덧셈')  print('5. 다항식 곱셈')  m = int(input('메뉴 선택 (종료시는 9) : '))  if m == 1:  print('다항식 입력 선택\n')  p = inputPoly()  print('다항식 리스트 : ', p,'\n')  elif m == 2:  print('다항식 출력 선택\n')  print('다항식 리스트 : ',p)  print('다항식 = ',end='')  printPoly(p)  print('\n')  elif m == 3:  print('다항식 계산 선택\n')  print('다항식 : ',end='')  printPoly(p)  print()  x = int(input('X = '))  print('계산 결과 : ',evalPoly(p, x))  print()  elif m == 4:  print('다항식 덧셈 선택\n')  print('첫 번째 다항식을 입력하세요')  A = inputPoly()  print('두 번째 다항식을 입력하세요')  B = inputPoly()  print('A = ',end='')  printPoly(A)  print()  print('B = ',end='')  printPoly(B)  print()  print('A + B = ',end='')  addPoly(A, B)  print('\n')  elif m == 5:  print('다항식 곱셈 선택\n')  print('첫 번째 다항식을 입력하세요')  A = inputPoly()  print('두 번째 다항식을 입력하세요')  B = inputPoly()  print('A = ',end='')  printPoly(A)  print()  print('B = ',end='')  printPoly(B)  print()  print('A \* B = ',end='')  multiplyPoly(A, B)  print('\n')  else:  if m != 9:  print('메뉴 선택 오류\n') |

