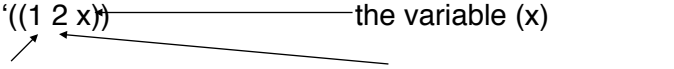
Test Case

username: jg495

Expression Formula
In my program, the polynomial is represented by this formula:
eg: x^2 = ‘((1 2 x))



constant 9 = ‘((9)) or ‘((9 0 0)) both is fine in my program

x^2y^3 = ‘((1 2 x 3 y))

x^2*y + y^2*z = ‘((1 2 x 1 y) (1 2 y 1 z))

Compiler: common lisp

poly1	poly2	test	except result	test result
test poly+ and poly-				
((2))	((1))	2+1	((3))	p
((10))	((-2))	10-2	((8))	p
((10))	((10))	10 - 10	((0))	p
((-10))	((-10))	-10-10	((-20))	p
((8))	((10))	8-10	((-2))	p
((0))	(())	0+0	((0))	p
((0))	((10))	0+10	((10))	p
((1 1 x))	((1 1 y))	x+y	((1 1 y) (1 1 x))	p
((1 1 x))	((1))	x-1	((1 1 x)(-1))	p
((1 1 x) (1))	((1 1 x) (1))	(x+1) - (x-1)	((0 1 x)(0))	p
((1 1 x) (1 1 y))	((1 1 x) (1 1 z))	(x+y) + (x+z)	((1 1 z) (2 1 x) (1 1 y))	p
((1 1 x) (1 1 y))	((1 1 x) (1 1 y))	(x+y) - (x+y)	((0 1 x) (0 1 y))	p
((1 1 x 1y))	((1 1 y 1 x))	xy-yx	((0 1 y) (0 1 x))	p
((1 1 x 1y))	((1 1 y 1 x))	xy+yx	((2 1 x 1 y))	p
((1 1 x 1 y) (1 3 x))	((1 1 x) (1 1 y 1 x))	(xy+x^3)+(x+yx)	((1 1 x) (2 1 x 1 y) (1 3 x))	p
((0))	((1 1 x 1 y))	0+xy	((1 1 x 1 y) (0))	p

poly1	poly2	test	except result	test result
((1 1 x 1 y 1 z))	((1 2 x))	xyz-x^2	((1 1 x 1 y 1 z)(-1 2 x))	p
((1 2 x))	((1 2 x))	x^2-x^2	(0 2 x)	p
((1 2 x 2 y))	(1 2 x 2 z)	(x^2y^2)-(x^2z^2)	((1 2 x 2 y) (-1 2 x 2 z))	p
((2 1 x))	(poly+ ((3 1 x) (2 1 y)))	2x-(poly+ 3x 2y)	((-2 1 y) (-1 1 x))	p
((1 1 x 1 y))	((1 1 y 1 1 x))	xy + yx	((2 1 x 1 y))	p
((1 1 x))	((1 1 x))	x-x	(0 1 x)	p
((1 1 x) (1 1 y) (1))	((2 1 x 1 y) (1 1 x) (1 1 z))	(x+y+1)+(2xy+x +z)	((2 1 X 1 Y) (1 1 Z) (2 1 X) (1 1 Y) (1))	p
((2 1 X 1 Y) (1 1 Z) (2 1 X) (1 1 Y) (1))	((2 1 x 1 y) (1 1 x) (1 1 z))	(2xy+2x+y+z +1) - (2xy+x +z)	((1 1 Y) (1) (0 1 X 1 Y) (1 1 X) (0 1 Z))	p
test poly*				
((2))	((3))	2*3	((6))	p
((1 1 x))	((1 1 x))	x*x	((1 2 x))	p
((1 1 x))	((1 1 y))	x*y	((1 1 y 1 x))	p
((1 1 x))	((1))	x*1	((1 1 x))	p
(poly+ ((1 1 x)) ((1)))	(poly+ ((1 1 y)) ((1)))	(x+1)*(y+1)	((1 1 X) (1 1 Y 1 X) (1) (1 1 Y))	p
(poly+ ((1 1 x)) ((1 1 y)))	(poly+ ((1 1 x)) ((1 1 y)))	(x+y)*(x+y)	((1 2 X) (1 2 Y) (2 1 X 1 Y))	p
(poly+ ((1 1 x)) (1 1 y)))	((1 1 x))	(x+y)*x	((1 2 X) (1 1 X 1 Y))	p
poly+ ((1 1 x)) ((1 1 y))	poly+ ((1 1 x)) ((1 1 z))	(x+y)*(x+z)	((1 1 Z 1 X) (1 2 X) (1 1 Z 1 Y) (1 1 X 1 Y))	p
((1 1 x 1 y))	((1 1 y 1 x))	xy*xy	((1 2 Y 2 X))	p
((2 1 x) (1))	((1 1 y)(1))	(2x+1)*(y+1)	((1 1 Y) (1) (2 1 Y 1 X) (2 1 X))	p
poly* ((1 1 x)) ((1 1 y))	poly+ ((1 1 x)) ((1 1 y))	(x*y)*(x+y)	((1 2 Y 1 X) (1 2 X 1 Y))	p
((0))	((1 1 x))	0*x	((0 1 x))	p
poly+ (poly* ((1 1 x)) ((1 1 y)) ((1)))	poly* ((1 1 y 1 z))	((x*y)+1)*yz	((1 2 Y 1 Z 1 X) (1 1 Y 1 Z))	p
((1 1 x) (1))	((1 1 x) (1))	(x+1)*(x+1)	((1) (1 2 x) (2 1 x))	p

The tests code and the results (you can use this code to test my code directly):

```
(poly+ ‘((2)) ‘((1)))
(poly+ ‘((10)) ‘((-2)))
(poly- ‘((10)) ‘((10)))
(poly+ ‘((-10)) ‘((-10)))
(poly- ‘((8)) ‘((10)))
(poly+ ‘((0)) ‘((0)))
(poly+ ‘((0)) ‘((10)))
(poly+ ‘((1 1 x)) ‘((1 1 y)))
(poly- ‘((1 1 x)) ‘((1)))
(poly- ‘((1 1 x) (1)) ‘((1 1 x) (1)))
(poly+ ‘((1 1 x) (1 1 y)) ‘((1 1 x)(1 1 z)))
(poly- ‘((1 1 x)(1 1 y)) ‘((1 1 x)(1 1 y)))
(poly- ‘((1 1 x)(1 1 y)) ‘((1 1 y)(1 1 x)))
(poly+ ‘((1 1 x 1 y)) ‘((1 1 y 1 x)))
(poly+ ‘((1 1 x 1 y)(1 3 x )) ‘((1 1 x)(1 1 y 1 x)))
(poly+ ‘((0)) ‘((1 1 x 1 y)))
(poly- ‘((1 1 x 1 y 1 z)) ‘((1 2 x)))
(poly- ‘((1 2 x)) ‘((1 2 x)))
(poly- ‘((1 2 x 2 y)) ‘((1 2 x 2 z)))
(poly- ‘((2 1 x)) (poly+ ‘((3 1 x)) ‘((2 1 y))))
(poly+ ‘((1 1 x 1 y)) ‘((1 1 y 1 x)))
(poly- '((1 1 x)) '((1 1 x)))
(poly+ '((1 1 x ) (1 1 y) (1)) '((2 1 x 1 y) (1 1 x) (1 1 z)))
(poly- '((2 1 X 1 Y) (1 1 Z) (2 1 X) (1 1 Y) (1)) '((2 1 x 1 y) (1 1 x) (1 1 z)))

(poly* '((2)) '((3)))
(poly* '((1 1 x)) '((1 1 x)))
(poly* '((1 1 x)) '((1 1 y)))
(poly* '((1 1 x)) '((1)))
(poly* (poly+ '((1 1 x)) '((1))) (poly+ '((1 1 y)) '((1))))
(poly* (poly+ '((1 1 x)) '((1 1 y))) (poly+ '((1 1 x)) '((1 1 y))))
(poly* (poly+ '((1 1 x)) '((1 1 y))) '((1 1 x)))
(poly* (poly+ '((1 1 x)) '((1 1 y))) (poly+ '((1 1 x)) '((1 1 z))))
(poly* '((1 1 x 1 y)) '((1 1 y 1 x)))
(poly* '((2 1 x)(1)) '((1 1 y)(1)))
(poly* (poly* '((1 1 x)) '((1 1 y))) (poly+ '((1 1 x)) '((1 1 y))))
(poly* '((0)) '((1 1 x)))
(poly* (poly+ (poly* '((1 1 x)) '((1 1 y))) '((1))) '((1 1 y 1 z)))
(poly* '((1 1 x)(1)) '((1 1 x)(1)))
```

```
CL-USER 7 > (poly+ '((2)) '((1)))
((3))

CL-USER 8 > (poly+ '((10)) '((-2)))
((8))

CL-USER 9 > (poly- '((10)) '((10)))
((0))

CL-USER 10 > (poly+ '((-10)) '((-10)))
((-20))

CL-USER 11 > (poly- '((8)) '((10)))
((-2))

CL-USER 12 > (poly+ '((0)) '((0)))
((0))

CL-USER 13 > (poly+ '((0)) '((10)))
((10))

CL-USER 14 > (poly+ '((1 1 x)) '((1 1 y)))
((1 1 Y) (1 1 X))

CL-USER 15 > (poly- '((1 1 x)) '((1)))
((1 1 X) (-1))

CL-USER 16 > (poly- '((1 1 x) (1)) '((1 1 x) (1)))
((0 1 X) (0))

CL-USER 17 > (poly+ '((1 1 x) (1 1 y)) '((1 1 x)(1 1 z)))
((1 1 Z) (2 1 X) (1 1 Y))

CL-USER 18 > (poly- '((1 1 x)(1 1 y)) '((1 1 x)(1 1 y)))
((0 1 X) (0 1 Y))

CL-USER 19 > (poly- '((1 1 x)(1 1 y)) '((1 1 y)(1 1 x)))
((0 1 Y) (0 1 X))

CL-USER 20 > (poly+ '((1 1 x 1 y)) '((1 1 y 1 x)))
((2 1 X 1 Y))

CL-USER 21 > (poly+ '((1 1 x 1 y)(1 3 x )) '((1 1 x)(1 1 y 1 x)))
((1 1 X) (2 1 X 1 Y) (1 3 X))

CL-USER 22 > (poly+ '((0)) '((1 1 x 1 y)))
((1 1 X 1 Y) (0))

CL-USER 23 > (poly- '((1 1 x 1 y 1 z)) '((1 2 x)))
((1 1 X 1 Y 1 Z) (-1 2 X))

CL-USER 24 > (poly- '((1 2 x)) '((1 2 x)))
((0 2 X))
```

```
CL-USER 25 > (poly- '((1 2 x 2 y)) '((1 2 x 2 z)))
((1 2 X 2 Y) (-1 2 X 2 Z))

CL-USER 26 > (poly- '((2 1 x)) (poly+ '((3 1 x)) '((2 1 y))))
((-2 1 Y) (-1 1 X))

CL-USER 27 > (poly+ '((1 1 x 1 y)) '((1 1 y 1 x)))
((2 1 X 1 Y))

CL-USER 28 > (poly- '((1 1 x)) '((1 1 x)))
((0 1 X))

CL-USER 29 > (poly+ '((1 1 x ) (1 1 y) (1)) '((2 1 x 1 y) (1 1 x) (1 1 z)))
((2 1 X 1 Y) (1 1 Z) (2 1 X) (1 1 Y) (1))

CL-USER 30 > (poly- '((2 1 X 1 Y) (1 1 Z) (2 1 X) (1 1 Y) (1)) '((2 1 x 1 y) (1 1 x) (1 1 z)))
((1 1 Y) (1) (0 1 X 1 Y) (1 1 X) (0 1 Z))

CL-USER 31 >
(poly* '((2)) '((3)))
((6))

CL-USER 32 > (poly* '((1 1 x)) '((1 1 x)))
((1 2 X))

CL-USER 33 > (poly* '((1 1 x)) '((1 1 y)))
((1 1 Y 1 X))

CL-USER 34 > (poly* '((1 1 x)) '((1)))
((1 1 X))

CL-USER 35 > (poly* (poly+ '((1 1 x)) '((1))) (poly+ '((1 1 y)) '((1))))
((1 1 X) (1 1 Y 1 X) (1) (1 1 Y))

CL-USER 36 > (poly* (poly+ '((1 1 x)) '((1 1 y))) (poly+ '((1 1 x)) '((1 1 y))))
((1 2 X) (1 2 Y) (2 1 X 1 Y))

CL-USER 37 > (poly* (poly+ '((1 1 x)) '((1 1 y))) '((1 1 x)))
((1 2 X) (1 1 X 1 Y))

CL-USER 38 > (poly* (poly+ '((1 1 x)) '((1 1 y))) (poly+ '((1 1 x)) '((1 1 z))))
((1 1 Z 1 X) (1 2 X) (1 1 Z 1 Y) (1 1 X 1 Y))

CL-USER 39 > (poly* '((1 1 x 1 y)) '((1 1 y 1 x)))
((1 2 Y 2 X))

CL-USER 40 > (poly* '((2 1 x)(1)) '((1 1 y)(1)))
((1 1 Y) (1) (2 1 Y 1 X) (2 1 X))

CL-USER 41 > (poly* (poly* '((1 1 x)) '((1 1 y))) (poly+ '((1 1 x)) '((1 1 y))))
((1 2 Y 1 X) (1 2 X 1 Y))
```

```
CL-USER 42 > (poly* '((0)) '((1 1 x)))
((0 1 X))

CL-USER 43 > (poly* (poly+ (poly* '((1 1 x)) '((1 1 y))) '((1 ))) '((1 1 y 1 z)))
((1 2 Y 1 Z 1 X) (1 1 Y 1 Z))
```

```
CL-USER 4 > (poly* '((1 1 x)(1)) '((1 1 x)(1)))
((1) (1 2 X) (2 1 X))
```