

```

void padd(int startA, int finishA, int startB, int finishB,
          int *startD, int *finishD);
{ /* A(x)와 B(x)를 더하여 D(x)를 생성한다. */
    float coefficient;
    *startD = avail;
    while (startA <= finishA && startB <= finishB)
        switch (COMPARE(terms[ startA] .expon,
                        terms[ startB] .expon)) {
            case -1: /* a의 expon이 b의 expon보다 작은 경우 */
                attach(terms[ startB] .coef, terms[ startB] .expon);
                startB++;
                break;
            case 0: /* 지수가 같은 경우 */
                coefficient = terms[ startA] .coef +
                             terms[ startB] .coef;
                if (coefficient)
                    attach(coefficient, terms[ startA] .expon);
                startA++;
                startB++;
                break;
            case 1: /* a의 expon이 b의 expon보다 큰 경우 */
                attach(terms[ startA] .coef, terms[ startA] .expon);
                startA++;
        }
    /* A(x)의 나머지 항들을 첨가한다. */
    for(; startA <= finishA; startA++)
        attach(terms[ startA] .coef, terms[ startA] .expon);
    /* B(x)의 나머지 항들을 첨가한다. */
    for(; startB <= finishB; startB++)
        attach(terms[ startB] .coef, terms[ startB] .expon);
    *finishD = avail-1;
}

```

**프로그램 2.6:** 두 다항식을 더하는 함수

---

```
void attach(float coefficient, int exponent)
{ /* 새로운 항을 다항식에 첨가한다. */
    if (avail >= MAX_TERMS) {
        fprintf(stderr, "다항식에 항이 너무 많다.");
        exit(1);
    }
    terms[avail].coef = coefficient;
    terms[avail++].expon = exponent;
}
```

---

**프로그램 2.7:** 새로운 항을 첨가하는 함수