We Know, by basic exponent law, $x^m \times x^n = x^{m+n}$

$\therefore$ The exponent of the term created by multiplying ~~an~~ two other terms having exponents terms[i].expon and terms[x].expon is terms[i].expon + terms[x].expon. It's easy to see that the coefficient of the product term is $\boxed{\text{terms[i].coef * terms[x].coef}}$. By eqn(2), there can be no other coefficients for this exponent which can be added up.

• On line 10, we input terms[i].coef * terms[x].coef as coefficient and terms[x].expon + terms[i].expon as exponent to the attach function. By correctness of the attach function, the new term is added to the global terms[] array. By eqn(2), we Know the original order is maintained.

• ~~To~~ Before the start of the next iteration i updates to x+1. It's easy to see now that the loop invariant is still maintained.

$\boxed{\text{Termination:}}$ The loop terminates when j = finishB+1. By the loop invariant, $\forall K$, startB $\leq K <$ finishB+1, the product terms terms[i].coef * terms[K].coef $* x^{(\text{terms[i].expon} + \text{terms[K].expon})}$ have been correctly computed and stored in the global terms[] array by the attach() function in their original order.

$\therefore$ The partial product term corresponding to terms[i] with polynomial B has been correctly computed and stored in the terms array. □

$\boxed{\text{Outer loop invariant:}}$ Just before the start of the ith iteration of the outer loop, for all K such that startA $\leq K < i$, the ~~partial~~ sum of each of the partial products corresponding to terms[K] with polynomial B has been correctly computed and stored in polynomial x. The polynomial x is represented in the terms array from index startx to finishx.