| 12. if (coefficient) |
|--|
| 12. if (coefficient) 13. attach/coefficient, terms [startA].expon); 14. startA++; startB++; |
| the oreary |
| 16. case 1: /*a expon > b expon 7. attach (terms [start] .coef, terms [start] - expon); |
| 7. attacrities is a |
| at A(2) |
| ofor(; startAn=finishA; startA++); startA].expon); |
| ofor (; start An = finish A; start A++) ofor (; start An = finish A; start A++) of of (; start A) . coef, terms [start A] . expon); of attach (terms [start A] . coef, terms [start A] . expon); of attach (terms [start A] . coef, terms [start A] . expon); |
| /*add in remaining terms of B(x) */ /*add in remaining terms of B(x) */ |
| /*add in remaining terms () cost, terms [start B]. expon); |
| /*add in remaining terms of the start b+) 25 or (; start b = finish b; start b+) 25 or (; start b = finish b = coef, terms [start b] . expon); 23. attach (terms [start b] . coef, terms [start b] . expon); * finish D = avail-1; 24. |
| * finish v= avaid 1 |
| Void attach/float coefficient, intexponent) 1. Void attach/float coefficient, intexponent */ |
| 1. Void attach (float coefficient, so 1) 2. 8/* add a new term to the polynomial (n"); 3. if (avail) = MAX_TERMS) { frints (stderp, is Too many terms in the polynomial (n")); f. exit (EXIT_FAILURE); 6. 17 and = coefficient; |
| 2. 1/4 add a MAX_TERMS) I siffavail) = MAX_TERMS) I terms in the polynomial In /; |
| 4. frints (stderp, a comas) |
| 5. QuitlExt Interest of the contract of the co |
| 6. Lerms [avail]. coef=coefficient; 7. terms [avail+]. expon=exponent; |
| 7. terms [availt]. expon = exponent; 9. } |
| 9. } |
| froof of Correctness The function is correct. |
| . Its easy to see that the attach function is correct. |
| Assumptions: Lacot payments the solumonial has terms stored |
| Assumptions: 1. Each polynomial is represented as a formal has terms stored descending order of exponents. That is, if the polynomial has terms stored descending order of exponents. That is, if the polynomial has terms stored from terms [startA] to terms [startA], then: terms [startA].expon terms [startA].expon |
| from forms starting to |
| · expon//terms[finishA]. expon |