Experiment 6
Addition Of Two Polynomials

Date: 21-09-2020

Aim: To receive two polynomials and print their sum

Data Structure Used: Arrays

Operation Used: Comparisons

Algorithm:

Input: Two polynomial, A and B in tuple format and 'a' denoting the number of tems in polynomial A and 'b' denoting the number of terms in polynomial 'B'

Output: Sum of the polynomial 'C'

Step 1 : Start
Step 2 : Receive two polynomial in tuple format
Step 3 : i ← 0 //Pointer to the polynomial A
Step 4 : j ← 0 //Pointer to the polynomial B
Step 5 : while i<a and j<b //a and b are the number of terms in A and B respectively
    Step 1 : if A[i][0] = B[j][0]
        Step 1: C[k][0] ← A[i][0]
        Step 2: C[k][1] ← A[i][1] +B[j][1]
        Step 3: i++
        Step 4: j++
        Step 5: k++
    Step 2: else if A[i][0] <B[j][0]
        Step 1: C[k][0] ← B[j][0]
        Step 2: C[k][1] ← B[j][1]
        Step 3: j++
        Step 6: k++
    Step 3: else if A[i][0] >B[j][0]
        Step 1: C[k][0] ← A[i][0]
        Step 2: C[k][1] ← A[i][1]
        Step 3: i++
        Step 4: k++
    Step 4: Endif
Step 6 : EndWhile
Step 7 : while i<a
    Step 1: C[k][0] ← A[i][0]
    Step 2: C[k][1] ← A[i][1]
    Step 3: i++
    Step 4: k++
Step 8: EndWhile
Step 9: while j<b
    Step 1: C[k][0] ← B[j][0]
    Step 2: C[k][1] ← B[j][1]
    Step 3: j++
    Step 6: k++
Step 10 : EndWhile
Step 11 : Stop

Description of the Algorithm:

The two polynomials are stored as two different 2-D arrays with the first column containing the powers of the polynomial (in descending order) and the second row containing the corresponding coefficients of the polynomial. Two pointers pointing to the two polynomials are created, if the powers pointed by the two polynomials are same then the coefficients are added and the result is pushed in the sum array, else the coefficient of the greater power is pushed into the sum array.

Result: the Program is successfully compiled and the desired output is obtained.

Program/ Source Code:

```
#include<stdio.h>
#include<stdlib.h>

/*  Input : 2 polynomials of the form
 *          a0*X^n + a1*X^n-1 + a2*X^n-2 ..... an*X^0
 *  Output: First polynomial the second polynomial and there sum
 */
/* Funtion to print the polynomials*/
void printPoly(int** a){
    int iterCount = a[0][0];
    int i;
    for(i = 1;i<iterCount;i++)
        printf("%d*X^%d + ",a[i][1],a[i][0]);
    printf("%d*X^%d\n",a[i][1],a[i][0]);
}

/* Funtion to convert the polynomial into tuple*/
int** createPolyFromString(char* s){
    int** a;
    int i,j;
    int maxPolySize = 10;

    int count = 0;
    int numberStack[10];
    int numberStackTop = -1;

    int number = 0;
    int negative = 0;

    //parsing the string

    a = (int**) malloc(maxPolySize*sizeof(int*));
    for(i = 0;i<maxPolySize;i++){
        a[i] = (int*)malloc(2*sizeof(int));
    }

    for(i = 0; s[i]!='\0'; i++){
        if(s[i] == '-'){
            negative = 1;
```

```
            i++;
        }

        if(s[i]>='0'&&s[i]<='9'){
            while((s[i]!= 'X'||s[i]!='x'||s[i]!=' '||s[i]!='^') &&
(s[i]>='0'&&s[i]<='9')){
                 // here s[i] will only be numbers
                number = number*10+(s[i]-'0');
                i++;
            }
            if(negative) numberStack[++numberStackTop] = -1*number;
            else numberStack[++numberStackTop] = number;


            negative = 0;
            number = 0;
        }
        if(s[i]=='+'||s[i]=='\0'){
                    count++;
                    a[count][0] = numberStack[numberStackTop--];
                    a[count][1] = numberStack[numberStackTop--];
        }

    }
    a[0][0] = count;

    return a;
}

/*Funtion to find the sum of the polynomials*/
int** sumOfPoly(int** a, int** b){
    int totalSize = a[0][0] + b[0][0]+2;
    int **c;
    int count = 0;
    c = (int**) malloc(totalSize*sizeof(int*));

    int i,j;
    for(i = 0;i<totalSize;i++){
        c[i] = (int*) malloc(2*sizeof(int));
    }

    i=1,j=1;
    while(i<=a[0][0]&&j<=b[0][0]){
        //If the powers are same then add the coefficients
        if(a[i][0]==b[j][0]){
            if(a[i][1]+b[j][1]==0){
                i++;j++;
                continue;
            }
            else{
                count++;
                c[count][0] = a[i][0];
                c[count][1] = a[i][1]+b[j][1];
                i++;j++;
            }
        }
```

```
        //If the powers arent same then push the one with the highest power into
polynomial c
        else if(a[i][0]<b[j][0]){
            count++;
            c[count][0] = b[j][0];
            c[count][1] = b[j][1];
            j++;
        }
        else if(b[j][0]<a[i][0]){
            count++;
            c[count][0] = a[i][0];
            c[count][1] = a[i][1];
            i++;
        }
    }

    /* If the while loop abve terminates prematurely i.e. after the elements of the
shorter of the two
       polynomial is added to the c polynomial*/

    while(i<=a[0][0]){
        count++;
        c[count][0] = a[i][0];
        c[count][1] = a[i][1];
        i++;
    }

    while(j<=b[0][0]){
        count++;
        c[count][0] = b[j][0];
        c[count][1] = b[j][1];
        j++;
    }

    c[0][0] = count;
    return c;
}
void main(){
    int** a;
    int** b;
    int** c;
    int strLength = 100;
    char* polyString = (char*) malloc(strLength*sizeof(char));

    /*Read the polynomials*/
        fflush(stdin);
        printf("Enter polynomial 1 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 .....
an*X^0 ");
        scanf("%[^\n]",polyString);
        scanf("%*c"); //remove the \n charecter from the input stream
        a  = createPolyFromString(polyString);
        free(polyString);

        fflush(stdin);
        fflush(stdout);
```

```
        polyString = (char*) malloc(strLength*sizeof(char));

        printf("Enter polynomial 2 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 .....
an*X^0 ");
        scanf("%[^\n]",polyString);
        b  = createPolyFromString(polyString);
        free(polyString);
    /*Finish reading Polynomials*/

    printf("\nPolynomial 1 is: ");
    printPoly(a);
    printf("\nPolynomial 2 is: ");
    printPoly(b);

    c = sumOfPoly(a,b); //Find the sum of the polynomials
    printf("\nSum is ");
    printPoly(c);
    free(a);
    free(b);
    free(c);
}
```

Sample Input/Output

Sample input 1:

```
100*X^10 + 29*X^5 + 10*X^0
21*X^9 + 1*X^5 + 3*X^3 + 2X^1
```

Sample output 1:

```
Enter polynomial 1 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 ..... an*X^0 -->
100*X^10 + 29*X^5 + 10*X^0
Enter polynomial 2 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 ..... an*X^0 -->
21*X^9 + 1*X^5 + 3*X^3 + 2X^1

Polynomial 1 is: 100*X^10 + 29*X^5 + 10*X^0

Polynomial 2 is: 21*X^9 + 1*X^5 + 3*X^3 + 2*X^1

Sum is 100*X^10 + 21*X^9 + 30*X^5 + 3*X^3 + 2*X^1 + 10*X^0
```

Sample input 2:

```
12*X^100 +12*X^1
13*X^101 + -12*X^100 + 1*X^2
```

Sample output 2:

```
Enter polynomial 1 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 ..... an*X^0 -->
12*X^100 +12*X^1
Enter polynomial 2 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 ..... an*X^0 -->
13*X^101 + -12*X^100 + 1*X^2

Polynomial 1 is: 12*X^100 + 12*X^1

Polynomial 2 is: 13*X^101 + -12*X^100 + 1*X^2
```

```
Sum is 13*X^101 + 1*X^2 + 12*X^1
```

Sample input 3:

```
-11*X^12 + 1*X^0
11*X^12 + 13*X^10 + 14*X^0
```

Sample output 3:

```
Enter polynomial 1 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 ..... an*X^0 --> -
11*X^12 + 1*X^0
Enter polynomial 2 in the form : a0*X^n + a1*X^n-1 + a2*X^n-2 ..... an*X^0 -->
11*X^12 + 13*X^10 + 14*X^0

Polynomial 1 is: -11*X^12 + 1*X^0

Polynomial 2 is: 11*X^12 + 13*X^10 + 14*X^0

Sum is 13*X^10 + 15*X^0
```