**Data Structure #1 Assignment**

2017320108 컴퓨터학과 고재영

Used language: C

Codes:

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

#define MAX_LEN 100                    // user can type input at most MAX_LEN

#define COMPARE(x, y) ( ((x) < (y)) ? -1 : ((x) == (y)) ? 0 : 1 )

typedef struct {    //design ADT for polynomial -> Polynomial
        float coef;
        int expon;
} Polynomial;

void showPoly(Polynomial p[], int term){      //this function shows Polynomial in console
        int k;
        for(k = 0; k < term; k++){
                if(p[k].coef == 0)
```

```
                continue;
        else if(p[k].expon == 0){
                printf(" + ");
                printf("%0.3f₩n", p[k].coef);
                break;
        }
        printf("%0.3fx^%d", p[k].coef, p[k].expon);
        if (p[k+1].expon != 0)
                printf(" + ");
    }
}


void addPoly(Polynomial p1[], Polynomial p2[], int t1, int t2, Polynomial p3[])        //add operation for polynomials
 {
    int i, j ,k;
    i=0;
    j=0;
    k=0;

    while(i < t1 && j < t2)
    {
```

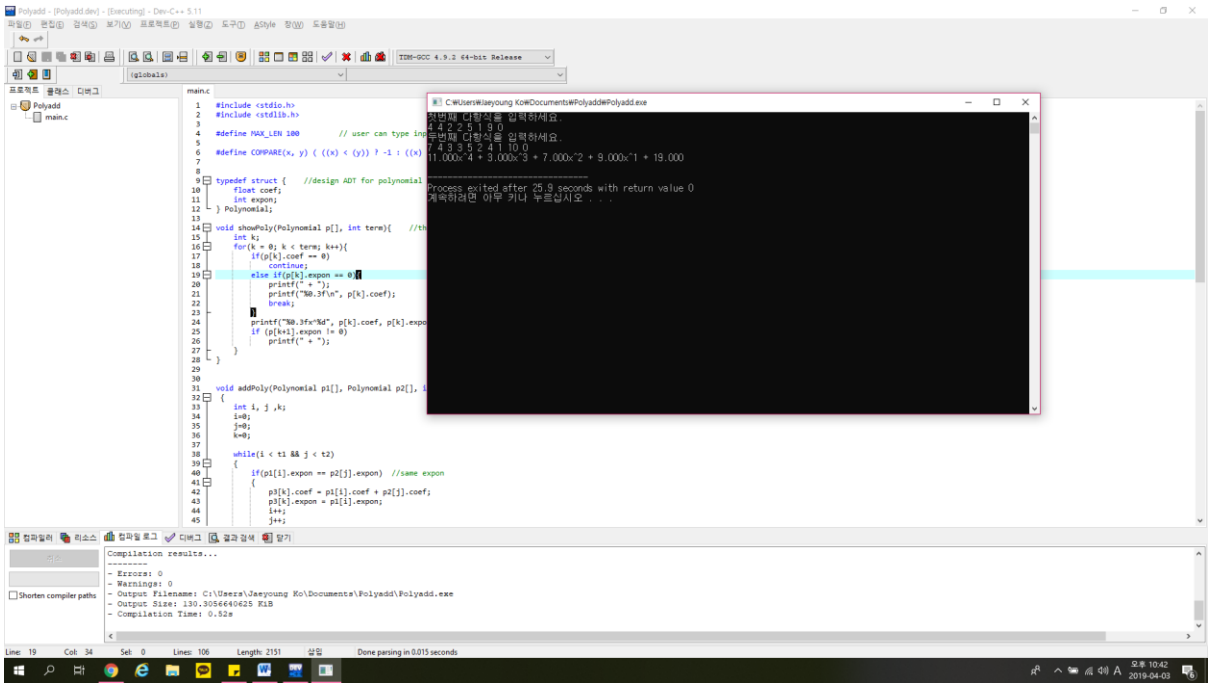
```
        if(p1[i].expon == p2[j].expon)        //same expon
        {
                p3[k].coef = p1[i].coef + p2[j].coef;
                p3[k].expon = p1[i].expon;
                i++;
                j++;
                k++;
        }
        else if(p1[i].expon > p2[j].expon)
        {
                p3[k].coef = p1[i].coef;
                p3[k].expon=p1[i].expon;
                i++;
                k++;
        }
        else
        {
                p3[k].coef = p2[j].coef;
                p3[k].expon = p2[j].expon;
                j++;
                k++;
        }
}
```

```
        /* for rest over terms of polynomial 1 */
        while(i < t1)
        {
                p3[k].coef = p1[i].coef;
                p3[k].expon = p1[i].expon;
                i++;
                k++;
        }
        /* for rest over terms of polynomial 2 */
        while(j < t2)
        {
                p3[k].coef = p2[j].coef;
                p3[k].expon = p2[j].expon;
                j++;
                k++;
        }
 }


int main() {
        Polynomial A[MAX_LEN], B[MAX_LEN], C[MAX_LEN];
        int i, j;
```

```
int ch;

printf("첫번째 다항식을 입력하세요.₩n");    //notify user to type first polynomial
for(i = 0; i < MAX_LEN; i++){                                    //user types first inputs
        scanf("%f %d", &A[i].coef, &A[i].expon);
        if (A[i].expon == 0)
                break;
}

showPoly(A, A[0].expon+1);

printf("두번째 다항식을 입력하세요.₩n");    //notify user to type second polynomial
for(j = 0; j < MAX_LEN; j++){                                    //user types second inputs
        scanf("%f %d", &B[j].coef, &B[j].expon);
        if (B[j].expon == 0)
                break;
}
showPoly(B, B[0].expon+1);

addPoly(A, B, A[0].expon+1, B[0].expon+1, C);
showPoly(C, C[0].expon+1);

return 0;
```

}

Screenshot:


Polyadd - [Polyadd.dev] - [Executing] - Dev-C++ 5.11
main.c
#include <stdio.h>
#include <stdlib.h>
#define MAX_LEN 100 // user can type inp
#define COMPARE(x, y) ( ((x) < (y)) ? -1 : ((x)
typedef struct { //design ADT for polynomial
float coef;
int expon;
} Polynomial;
void showPoly(Polynomial p[], int term){ //th
int k;
for(k = 0; k < term; k++){
if(p[k].coef == 0)
continue;
else if(p[k].expon == 0){
printf(" + ");
printf("%0.3f\n", p[k].coef);
break;
}
printf("%0.3fx^%d", p[k].coef, p[k].expo
if (p[k+1].expon != 0)
printf(" + ");
}
}
void addPoly(Polynomial p1[], Polynomial p2[],
{
int i, j ,k;
i=0;
j=0;
k=0;
while(i < t1 && j < t2)
{
if(p1[i].expon == p2[j].expon) //same expon
{
p3[k].coef = p1[i].coef + p2[j].coef;
p3[k].expon = p1[i].expon;
i++;
j++;
C:\Users\Jaeyoung Ko\Documents\Polyadd\Polyadd.exe
첫번째 다항식을 입력하세요.
4 4 2 2 5 1 8 0
두번째 다항식을 입력하세요.
7 4 3 3 5 2 4 1 10 0
11.000x^4 + 3.000x^3 + 7.000x^2 + 9.000x^1 + 19.000
Process exited after 25.9 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Jaeyoung Ko\Documents\Polyadd\Polyadd.exe
- Output Size: 130.3056640625 KiB
- Compilation Time: 0.52s
Done parsing in 0.015 seconds