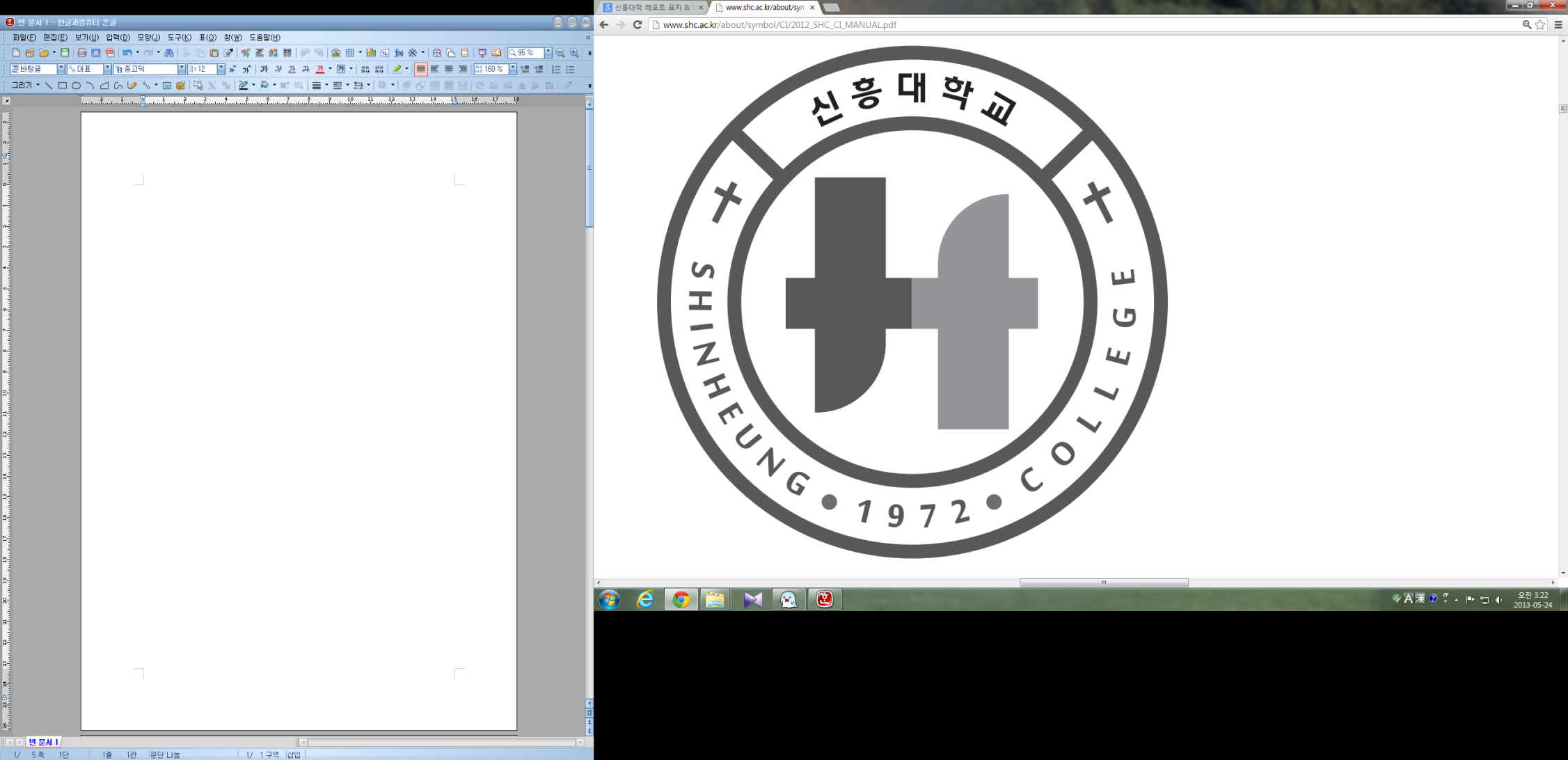
REPORT

다항식의 덧셈 프로그램



과 목 명 : 자료구조

담당교수 :김 창희 교수님

제출일자 :2013. 10. 02

학 과 : 소프트웨어개발전공

학 번 :2013301165

이 름 :정 연주

**1. 프로그램 소개**

선형리스트를 사용하여 다항식의 연산을 처리하는 프로그램이다. 배열과 구조체, 함수를 사용한다.

**2. 입출력 설계**

|  |
| --- |
| addPoly(A, B)  // 주어진두다항식 A와 B를더하여결과다항식 C를반환하는알고리즘  C <- zeroP();  While (not isZeroP(A) and not isZeroP(B)) do {  case {  maxExp(A) < maxExp(B) :  C <- addTerm(C, coef(B, maxExp(B)), maxExp(B));  B <- delTerm(B, maxExp(B));  maxExp(A) = maxExp(B) :  sum <- coef(A, maxExp(A)) + coef(B, maxExp(B));  if(sum ≠ 0) then C <- addTerm(C, sum, maxExp(A))  A<- delTerm(A, maxExp(A));  B<- delTerm(B, maxExp(B));  }  }  if (not isZero(A)) then A의나머지항들을 C에복사  else if (not isZeroP(B)) then B의나머지항들을 C에복사  return C;  End addPoly() |

두 다항식 A,B의 항을 하나씩 검사하여 그 지수가 같으면 계수를 더해서 만든 항을 다항식 C에 추가 addTerm() 하고 지수가 서로 다른 경우에는 둘 중에서 지수가 큰 항을 다항식 C에 추가하는 과정을 마지막 항까지 반복하여 덧셈결과 다항식 C를 완성한다.

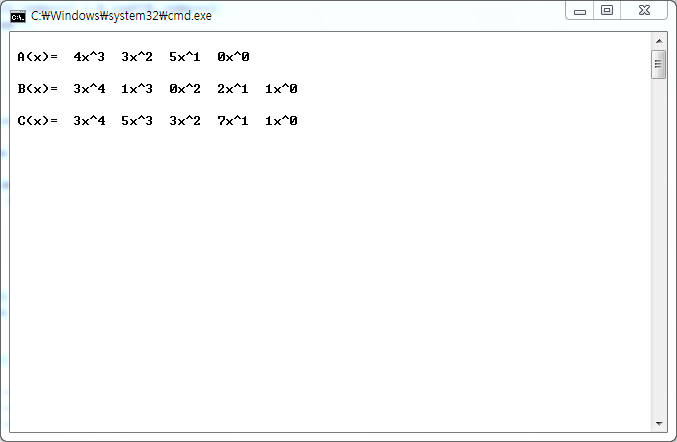
**3. 함수 정의**

|  |  |
| --- | --- |
| 함수명 | 설명 |
| Home(); | print menu and return menu value |
| addPoly(); | adding polynomial A and B to return C that result of adding |
| printPoly(); | print polynomial |
| main(); | main function |

**4. Source Code**

|  |
| --- |
| /\*다항식덧셈프로그램\*/  #include <stdio.h>  #define MAX(a,b) ((a>b)?a:b)  #define MAX\_DEGREE 50  /\*구조체 polynomial선언\*/  typedef struct{  int degree;  float coef[MAX\_DEGREE];  } polynomial;  polynomial addPoly(polynomial A, polynomial B){  polynomial C;  int A\_index=0, B\_index=0, C\_index=0;  int A\_degree=A.degree, B\_degree=B.degree;  C.degree=MAX(A.degree, B.degree);  while(A\_index<=A.degree && B\_index<=B.degree){  if(A\_degree > B\_degree){  C.coef[C\_index++] = A.coef[A\_index++];  A\_degree--;  }  else if(A\_degree == B\_degree){  C.coef[C\_index++] = A.coef[A\_index++]+B.coef[B\_index++];  A\_degree--;  B\_degree--;  }  else{  C.coef[C\_index++] = B.coef[B\_index++];  B\_degree--;  }  }  return C;  }  void printPoly(polynomial P){ //polynomial 출력  int i, degree;  degree=P.degree;  for(i=0; i<=P.degree; i++)  printf("%3.0fx^%d", P.coef[i], degree--);  printf("\n");  }  void main(){  polynomial A={3, {4,3,5,0}}; //{최고차항, {계수}}  polynomial B={4, {3,1,0,2,1}}; //{최고차항, {계수}}  polynomial C;  C= addPoly(A,B); //함수 addPoly의결과 C에반환  printf("\n A(x)="); printPoly(A);  printf("\n B(x)="); printPoly(B);  printf("\n C(x)="); printPoly(C);  getchar();  } |

**5. Result Screen**



**6. 느낀점**

선형리스트로 다항식의 덧셈을 하려고 구상하니 배열의 크기를 미리 정해놓고 입력을 받고 싶었지만 그렇게 하지 않고 최고차항을 먼저 입력 받은 후 각각의 항의 계수를 입력 받는 형식으로 구상했지만 실패했다. 실패한 코드. C언어를 더 열심히 공부해야 겠다

|  |
| --- |
| /\* 프로그램명 : 다항식의 덧셈 프로그램 Polynomial  작성자 : 정 연주 작성일 : 2013.10.01 \*/  #include<stdio.h>  #include<stdlib.h>  #define MAX(a,b) ((a>b)?a:b) // a>b가 참이면 a반환, 거짓이면 b반환  #define MAX\_DEGREE 50  void printP();  /\*구조체 선언\*/  typedef struct{  int degree; //항의 차수를 저장 변수  float coef[MAX\_DEGREE]; //항의 계수를 저장 1차원 배열  } polynomial;  polynomial addPoly(polynomial A, polynomial B) // add polynomial  {  polynomial C;  int A\_index=0, B\_index=0, C\_index=0;  int A\_degree=A.degree, B\_degree=B.degree;  C.degree=MAX(A.degree,B.degree);  while(A\_index<=A.degree && B\_index<=B.degree){  if(A\_degree > B\_degree){  C.coef[C\_index++] = A.coef[A\_index++];  A\_degree--;  }  else if(A\_degree == B\_degree){  C.coef[C\_index++]=A.coef[A\_index++]+B.coef[B\_index++];  A\_degree--;  B\_degree--;  }  else{  C.coef[C\_index++] = B.coef[B\_index++];  B\_degree--;  }  }  return C;  }  void printPoly(polynomial P)  {  int i, degree;  degree=P.degree;  for(i=0; i<=P.degree; i++)  printf("%3.0fx^%d",P.coef[i], degree--);  printf("\n");  }  int Home(){ // Start Menu  int val;  printf(" \*\*\* 다항식 덧셈 프로그램 \*\*\* \n");  printf("1. 덧셈하기 \n");  printf("2. 종료 \n");  printf(">> ");  scanf("%d", &val);  return val;  }  polynomial inputA(polynomial A){ // input A of coefficient and exponent  int i;  int A\_index=0;  //system("cls");  printf("첫번째 식(A)의 최고차항 입력 : ");  scanf("%d", &A.degree);  printf("\t첫번째 식(A)의 계수 입력 : ");  for (i=A.degree; i==0; i--){  scanf("%d", A.coef[A\_index]);  printf("\n");  }  polynomial A = {A.degree, {A.coef[A\_index]}};  printf("\n\n");  return A;  }  polynomial inputB(polynomial B){ // input B of coefficient and exponent  int i;  int B\_index=0;  printf("두번째 식(B)의 최고차항 입력 : ");  scanf("%d", &B.degree);  printf("\t두번째 식(B)의 계수 입력 : ");  for (i=B.degree; i==0; i--){  scanf("%d", B.coef[B\_index]);  printf("\n");  }  polynomial B = {B.degree, {B.coef[B\_index]}};  return B;  }  void main()  {  int val;  polynomial A;  polynomial B;  polynomial C;  val = Home();  switch (val) {  case 1:  inputA(A);  inputB(B);  C=addPoly(A, B);  printf("\n A(x)="); printPoly(A);  printf("\n B(x)="); printPoly(B);  printf("\n C(x)="); printPoly(C);  break;  case 2:  exit(1);  break;  default:  getchar();  break;  }    getchar(); \*/  } |