|  |  |  |  |
| --- | --- | --- | --- |
| 2022/2 『자료구조』실습 보고서 | | | |
| 제목 | 3장 실습( O ) 과제( ) | 제출일자 | 2022.   09.  28 . |
| 학번 | 201911608 | 이름 | 김지환 |

|  |
| --- |
| 실습 5. 프로그램 3.2  #include <stdio.h>  #define MAX(a,b) (a>b?a:b)  #define MAX\_DEGREE 101  typedef struct {  int degree;  float coef[MAX\_DEGREE];  } polynomial;  polynomial poly\_add1(polynomial A, polynomial B) {  polynomial C;  int Apos = 0, Bpos = 0, Cpos = 0;  int degree\_a = A.degree;  int degree\_b = B.degree;  C.degree = MAX(A.degree, B.degree);  while (Apos <= A.degree && Bpos <= B.degree) {  if (degree\_a > degree\_b) {  C.coef[Cpos++] = A.coef[Apos++];  degree\_a--;  }  else if (degree\_a == degree\_b) {  C.coef[Cpos++] = A.coef[Apos++] + B.coef[Bpos++];  degree\_a--; degree\_b--;  }  else {  C.coef[Cpos++] = B.coef[Bpos++];  degree\_b;  }  }  return C;  }  void print\_poly(polynomial p) {  for (int i = p.degree; i > 0; i--)  printf("%3.1fx^%d + ", p.coef[p.degree - i], i);  printf("%3.1f \n", p.coef[p.degree]);  }  int main() {  polynomial a = { 5, {3,6,0,0,0,10} };  polynomial b = { 4, {7,0,5,0,1} };  polynomial c;    print\_poly(a);  print\_poly(b);  c = poly\_add1(a, b);  printf("-------------------------------------------------------\n");  print\_poly(c);  return 0;  }  실행결과    교고서의 결과와 동일하다. |
| 실습 6. 프로그램 3.2 ( 아래 ) 도전문제  #include <stdio.h>  #define MAX(a,b) (a>b?a:b)  #define MAX\_DEGREE 101  typedef struct {  int degree;  float coef[MAX\_DEGREE];  } polynomial;  polynomial poly\_add1(polynomial A, polynomial B) {  polynomial C;  int Apos = 0, Bpos = 0, Cpos = 0;  int degree\_a = A.degree;  int degree\_b = B.degree;  C.degree = MAX(A.degree, B.degree);  while (Apos <= A.degree && Bpos <= B.degree) {  if (degree\_a > degree\_b) {  C.coef[Cpos++] = A.coef[Apos++];  degree\_a--;  }  else if (degree\_a == degree\_b) {  C.coef[Cpos++] = A.coef[Apos++] + B.coef[Bpos++];  degree\_a--; degree\_b--;  }  else {  C.coef[Cpos++] = B.coef[Bpos++];  degree\_b;  }  }  return C;  }  void print\_poly(polynomial p) {  for (int i = p.degree; i > 0; i--) {  if (p.coef[p.degree - i] == 0) continue;  printf("%3.1fx^%d + ", p.coef[p.degree - i], i);  }  printf("%3.1f \n", p.coef[p.degree]);  }  int main() {  polynomial a = { 3, {1,0,2,3} };  polynomial b = { 3, {-1,0,4,-1} };  polynomial c;  print\_poly(a);  print\_poly(b);  c = poly\_add1(a, b);  printf("-------------------------------------------------------\n");  print\_poly(c);  return 0;  }  실행결과    -> 도전문제에서 요구하는 문제점을 출력 함수에서 값이 0일 경우 continue를 사용해 해결하였다. |
| 실습 7. 프로그램 3.3  #include <stdio.h>  #include <stdlib.h>  #define MAX\_TERMS 101  typedef struct {  float coef;  int expon;  } polynomial;  polynomial terms[MAX\_TERMS] = { { 8,3 },{ 7,1 },{ 1,0 },{ 10,3 },{ 3,2 },{ 1,0 } };  int avail = 6;  char compare(int a, int b)  {  if (a > b) return '>';  else if (a == b) return '=';  else return '<';  }  void attach(float coef, int expon)  {  if (avail > MAX\_TERMS) {  fprintf(stderr, "항의 개수가 너무 많음\n");  exit(1);  }  terms[avail].coef = coef;  terms[avail].expon = expon;  avail++;  }  void poly\_add2(int As, int Ae, int Bs, int Be, int\* Cs, int\* Ce) {  float tempcoef;  \*Cs = avail;  while (As <= Ae && Bs <= Be) {  switch (compare(terms[As].expon, terms[Bs].expon)) {  case '>':  attach(terms[As].coef, terms[As].expon);  As++; break;  case '=':  tempcoef = terms[As].coef + terms[Bs].coef;  if (tempcoef)  attach(tempcoef, terms[As].expon);  As++;Bs++; break;  case '<':  attach(terms[Bs].coef, terms[Bs].expon);  Bs++; break;  }  }  for (; As <= Ae; As++) attach(terms[As].coef, terms[As].expon);  for (; Bs <= Be; Bs++) attach(terms[Bs].coef, terms[Bs].expon);  \*Ce = avail - 1;  }  void print\_poly(int s, int e) {  for (int i = s; i < e;i++)  printf("%3.1fx^%d + ", terms[i].coef, terms[i].expon);  printf("%3.1fx^%d\n", terms[e].coef, terms[e].expon);  }  int main() {  int As = 0, Ae = 2, Bs = 3, Be = 5, Cs, Ce;  poly\_add2(As, Ae, Bs, Be, &Cs, &Ce);  print\_poly(As, Ae);  print\_poly(Bs, Be);  printf("-----------------------------------\n");  print\_poly(Cs, Ce);  return 0;  }  실행결과    -> 교재 실행결과와 동일하다 |
| 실습 8 . 3-3 (QUIZ)  01 다항식 6x^3 + 8x^2 +0을 첫 번째 방법으로 표현하여 보라.  -> 실습 5번에서 main 함수를 이렇게 작성하면 된다.  int main() {  polynomial a = { 4, {6,8,0,9} };  print\_poly(a);  return 0;  }  실행결과    02. 다항식 6x^3 + 8x^2 +0을 두 번째 방법으로 표현하여 보라.  -> 실습 7번에서 전역배열인 terms와 main 함수를 이렇게 작성하면 된다.  polynomial terms[MAX\_TERMS] = { { 6, 3 }, { 8, 2 }, { 9,0 } };  int main() {  int As = 0, Ae = 2;  print\_poly(As, Ae);  return 0;  }  실행결과    03. 첫 번째 방법과 두 번째 방법으로 각각 다항식의 뺄셈을 구현하려면 덧셈 코드의 어떤 부분을 변경하면 되는가?  첫 번째 방법, poly\_add1() 함수 내 polynomial C 변수에 치환 하는 부분을 + 에서 - 로 수정한다.  C.coef[Cpos++] = A.coef[Apos++] - B.coef[Bpos++];  그 후, main 함수를 아래와 같이 수정한다.  int main() {  polynomial a = { 3, {6,8,0,9} };  polynomial b = { 3, {4,4,0,5} };  polynomial c = poly\_add1(a, b);  print\_poly(a);  print\_poly(b);  printf("===============================\n");  print\_poly(c);  return 0;  }  실행결과    두 번째 방법, poly\_add2() 함수 내 switch-case 문에서 변수 tempcoef에 치환하는 부분을 + 에서 -로 수정한다.  tempcoef = terms[As].coef - terms[Bs].coef;  그 후 , 전역 변수 terms와 main 함수를 아래와 같이 수정한다.  polynomial terms[MAX\_TERMS] = { { 6, 3 }, { 8, 2 }, { 9,0 }, { 4, 3 }, { 4, 2 }, { 5, 0 } };  int main() {  int As = 0, Ae = 2, Bs = 3, Be = 5, Cs, Ce;  poly\_add2(As, Ae, Bs, Be, &Cs, &Ce);  print\_poly(As, Ae);  print\_poly(Bs, Be);  printf("===============================\n");  print\_poly(Cs, Ce);  return 0;  }  실행 결과    -> 첫 번째 방법의 값과 두 번째 방법의 값이 같다.  실습 9. 프로그램 3.5  #include <stdio.h>  #include <stdlib.h>  #define MAX\_TERMS 100  typedef struct {  int row;  int col;  int value;  } element;  typedef struct SparseMatrix {  element data[MAX\_TERMS];  int rows;  int cols;  int terms;  } SparseMatrix;  SparseMatrix matrix\_transpose2(SparseMatrix a) {  SparseMatrix b;  int bindex;  b.rows = a.rows;  b.cols = a.cols;  b.terms = a.terms;  if (a.terms > 0) {  bindex = 0;  for (int c = 0; c < a.cols; c++)  for (int i = 0; i < a.terms; i++) {  if (a.data[i].col == c) {  b.data[bindex].row = a.data[i].col;  b.data[bindex].col = a.data[i].row;  b.data[bindex].value = a.data[i].value;  bindex++;  }  }  }  return b;  }  void matrix\_print(SparseMatrix a) {  printf("====================\n");  for (int i = 0; i < a.terms; i++)  printf("(%d, %d, %d) \n", a.data[i].row, a.data[i].col, a.data[i].value);  printf("====================\n");  }  int main() {  SparseMatrix m = {  { { 0, 3, 7 },{ 1, 0, 9 },{ 1, 5, 8 },{ 3, 0, 6 },{ 3, 1, 5 },{ 4, 5, 1 },{ 5, 2, 2 } }, 6, 6, 7  };  SparseMatrix result;  result = matrix\_transpose2(m);  matrix\_print(result);  return 0;  }  실행결과 |
| 실습 10. 프로그램 3.6  #include <stdio.h>  void swap(int\* px, int\* py) {  int tmp;  tmp = \*px;  \*px = \*py;  \*py = tmp;  }  int main() {  int a = 1, b = 2;  printf("swap을 호출하기 전: a=%d, b=%d\n", a, b);  swap(&a, &b);  printf("swap을 호출한 다음: a=%d, b=%d\n", a, b);  return 0;  }  실행결과 |