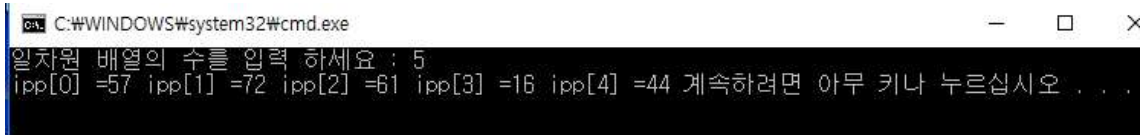


자료구조응용

03. 배열, 구조체, 다항식더하기

1. 입력 받은 일차원 배열의 원소 개수를 함수 `make1dArray()`에 전달하면, 함수 `make1dArray()`는 전달 받은 원소 갯수 만큼 동적으로 메모리를 할당 받아 램덤으로 100보다 적은 값으로 메모리를 초기화 시킨 후, 주소값을 반환한다. (즉 함수 호출로 1차원 배열 만들기)

```
int main()
{
    int *ipp;
    int i, num;
    .....
    ipp=make1dArray(num);
    for(i=0; i<num; i++)
        printf("ipp[%d] =%d ", i, ipp[i]);
}
```



2. 입력 받은 이차원 배열의 행의 수와 열의 수를 함수 `make2dArray()`에 전달하면, 함수 `make2dArray()`는 전달 받은 원소 갯수 만큼 동적으로 메모리를 할당 받아 램덤으로 100보다 적은 값으로 메모리를 초기화 시킨 후, 주소값을 반환한다. (즉 함수 호출로 2차원 배열 만들기) 프로그램을 작성하고 반환 직전에 변수값을 디버깅하세요. (Autos, Watch 창 사용)

```
int main()
{
    int **ipp;
    int i, j, rows, cols;
    printf("이차원 배열의 행과 열의 수를 입력 하세요 : ");
    scanf("%d %d",&rows, &cols);
    ipp=make2dArray(rows, cols);
    for(i=0; i<rows; i++)
        for(j=0; j<cols; j++)
            printf("ipp[%d] =%d ", i, ipp[i][j]);
    return 0; // break point 설정
}
```

Autos		
Name	Value	Type
i	2	int
ipp	0x00b94200	int **
	0x00b94258	int *
	5	int
ipp[i]	0xcdcdcdcd	int *
	CXX0030: Error: expression cannot be evaluated	
j	3	int

Watch 1		
Name	Value	Type
ipp[0][1]	29	int
ipp[0][2]	24	int
ipp[1][0]	88	int

```

D:\자료구조응용\WDS 03\Debug\W01-2.exe
이차원 배열의 행과 열의 수를 입력 하세요 : 2 3
ipp[0] =5 ipp[0] =29 ipp[0] =24 ipp[1] =88 ipp[1] =72 ipp[1] =93

```

Microsoft Visual Studio

3. 이차원 배열의 합을 구하는 프로그램이다. [2차원 배열을 함수로 전달] 정적배열, 동적배열을 함수로 전달하는 방법을 비교하는 프로그램이다. 함수의 파라미터 타입에 주의하면서 프로그램을 작성하라. 디버깅 창을 출력하고, 동적배열을 sumAry2D_f1의 인자로 전달할 수 없는 이유를 쓰시오.

// 정적배열 전달, 첫 번째 파라미터 타입은 모두 int [3]* --> 디버거로 확인하여 출력(디버깅 Autos 창에서 확인)

```
int sumAry2D_f1( int ary[][3], int rows, int cols ); // 배열 파라미터
int sumAry2D_f2( int (*ary)[3], int rows, int cols ); // 배열 포인터
int sumAry2D_f3( int ary[2][3], int rows, int cols );
```

// 동적배열 전달

```
int sumAry2D_f4( int **ary, int rows, int cols ); // int **
int sumAry2D_f5( int ***ary, int rows, int cols ); // int ***
```

```
void freeAry2D(int **ary, int rows);
```

```
int main(void)
{
    // 정적배열(2행3열)
    int ary2D[ ][3] = { {1, 2, 3},
                        {4, 5, 6} };

    // 동적배열(2행3열)
    int r, c;
    int **ary = (int **) malloc(sizeof(int*) * 2);
    for ( r = 0; r < 2; r++)
        ary[r] = (int *) malloc( sizeof(int) * 3);

    for ( r = 0; r < 2; r++)
        for ( c = 0; c < 3; c++)
            ary[r][c] = r+c;

    // 정적배열
    printf("sumAry2D_f1() %d\n", sumAry2D_f1(ary2D, 2, 3));
    printf("sumAry2D_f2() %d\n", sumAry2D_f2(ary2D, 2, 3));
    printf("sumAry2D_f3() %d\n", sumAry2D_f3(ary2D, 2, 3));

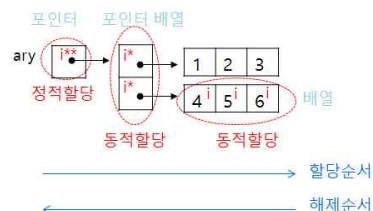
    // 동적배열
    printf("sumAry2D_f4() %d\n", sumAry2D_f4(ary, 2, 3));
    printf("sumAry2D_f5() %d\n", sumAry2D_f5(&ary, 2, 3));

    // 동적배열을 f1, f2, f3로 전달할 수 있을까? 테스트해 보라!
    //printf("sumAry2D_f1~f3() %d\n", sumAry2D_f1(ary, 2, 3));

    // 정적배열을 f4, f5로 전달할 수 있을까? 테스트해 보라!
    //printf("sumAry2D_f4~f5() %d\n", sumAry2D_f4(ary2D, 2, 3));

    freeAry2D(ary, 2);

    return 0;
}
```



[실행예]

```
C:\Windows\system32\cmd.e...
sumary2D_f1< 21
sumary2D_f2< 21
sumary2D_f3< 21
sumary2D_f4< 9
sumary2D_f5< 9
2d array - free!
계속하려면 아무 키나 누르십시오 . . .
```

[디버깅 Autos창]

Name	Value	Type
array	0x008fbee8	int [3]
cols	3	int
i	-858993460	int
j	-858993460	int
rows	2	int
sum	-858993460	int

4. [구조체] 다음 코드를 이용하여, 실행 예와 같은 결과를 나타내는 프로그램을 작성하라.
단 humansEqual 함수를 다음과 같이 수정하여 사용해야 한다.

수정된 함수원형 : `int humansEqual(humanBeing *person1, humanBeing *person2);`

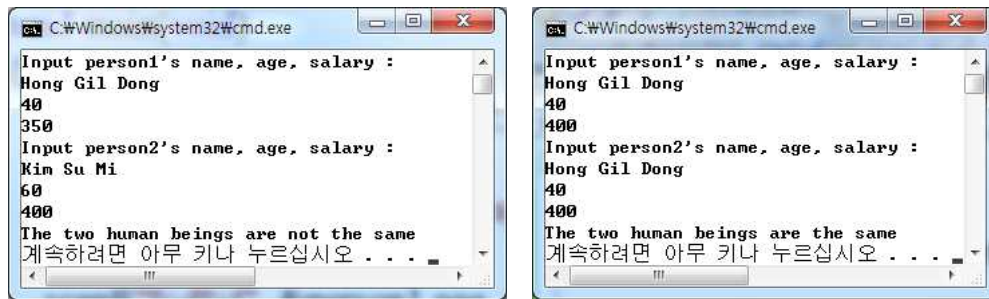
```
#define FALSE 0
#define TRUE 1

if (humansEqual(person1, person2))
    printf("The two human beings are the same\n");
else
    printf("The two human beings are not the same\n");

int humansEqual(humanBeing person1,
                humanBeing person2)
{
    /* return TRUE if person1 and person2 are the same human
       being otherwise return FALSE */
    if (strcmp(person1.name, person2.name))
        return FALSE;
    if (person1.age != person2.age)
        return FALSE;
    if (person1.salary != person2.salary)
        return FALSE;
    return TRUE;
}
```

Program 2.4: Function to check equality of structures

[실행예 및 조건]



※ 공백을 포함한 이름을 입력하여야 함

5. 다항식을 아래와 같이 표현 하는 방식을 이용하여, 두 다항식을 입력받아 더한 후 그 결과를 출력하는 프로그램을 작성하라.

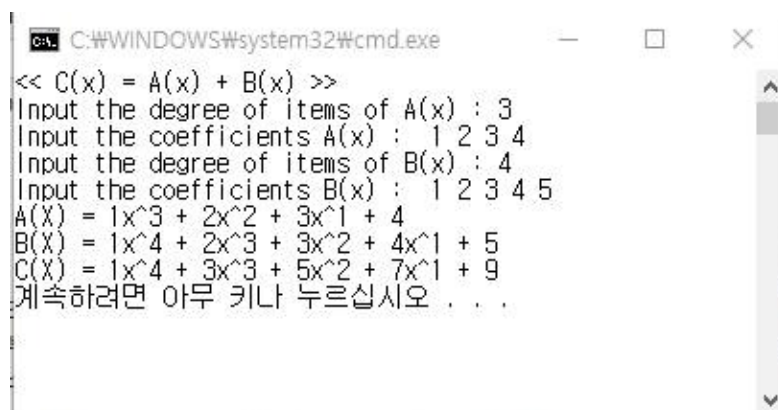
```
#define MAX-DEGREE 101 /*Max degree of polynomial+1*/
typedef struct {
    int degree;
    float coef[MAX-DEGREE];
} polynomial;
```

[프로그램설명]

- ① 각 다항식을 사용자로부터 키보드 입력을 받음
- ② 다항식을 출력하는 함수를 정의하여 사용하기

```
void poly_print(char [], polynomial);
```

[프로그램설명]



6. 다항식을 다음 같이 표현 하는 방식을 이용하여, 두 다항식을 입력받아 더한 후 그 결과를 출력하는 프로그램을 작성하라.

```
#define MAX_TERMS 100 /*size of terms array*/
typedef struct {
    float coef;
    int expon;
} term;
term terms[MAX_TERMS];
```

- ① 각 다항식을 사용자로부터 키보드 입력을 받음
- ② 교재의 `padd`, `attach` 함수는 수정 없이 그대로 사용하기
- ③ 다항식을 출력하는 함수를 정의하여 사용하기

```
void padd(int startA, int finishA, int startB, int finishB,
          int *startD, int *finishD)
/* add A(x) and B(x) to obtain D(x) */
float coefficient;
*startD = avail;
while (startA <= finishA && startB <= finishB)
    switch (COMPARE(terms[startA].expon,
                   terms[startB].expon)) {
        case -1: /* a expon < b expon */
            attach(terms[startB].coef, terms[startB].expon);
            startB++;
            break;
        case 0: /* equal exponents */
            coefficient = terms[startA].coef +
                          terms[startB].coef;
            if (coefficient)
                attach(coefficient, terms[startA].expon);
            startA++;
            startB++;
            break;
        case 1: /* a expon > b expon */
            attach(terms[startA].coef, terms[startA].expon);
            startA++;
    }
/* add in remaining terms of A(x) */
for(; startA <= finishA; startA++)
    attach(terms[startA].coef, terms[startA].expon);
/* add in remaining terms of B(x) */
for(; startB <= finishB; startB++)
    attach(terms[startB].coef, terms[startB].expon);
*finishD = avail-1;
}
```

Program 2.6: Function to add two polynomials

```

void attach(float coefficient, int exponent)
{
    /* add a new term to the polynomial */
    if (avail >= MAX_TERMS) {
        fprintf(stderr, "Too many terms in the polynomial\n");
        exit(EXIT_FAILURE);
    }
    terms[avail].coef = coefficient;
    terms[avail++].expon = exponent;
}

```

Program 2.7: Function to add a new term

[실행 예]

```

C:\WINDOWS\system32\cmd.exe
<< D(x) = A(x) + B(x) >>
Input the number of items of A(x) : 5
Input the number of items of B(x) : 3

input in descending order
coefficient and exponent of A(x)=10x^2+8x^2+3 (10 3, 8 2, 3 0) : 11 9, 9 5, 3 3, 7 2, 9 0
coefficient and exponent of B(x)=10x^2+8x^2+3 (10 3, 8 2, 3 0) : 10 9, 5 3, 5 2

A(x) = 11x^9 + 9x^5 + 3x^3 + 7x^2 + 9
B(x) = 10x^9 + 5x^3 + 5x^2
D(x) = 21x^9 + 9x^5 + 8x^3 + 12x^2 + 9
계속하려면 아무 키나 누르십시오 . . .

```

■ 제출 형식

- 공학인증 시스템(ABEEK)에 과제를 올릴 때 제목:
- 1차 제출: 학번_이름_DS_03(1), 2차 제출: 학번_이름_DS_03(2)
- 솔루션 이름 : DS_03
- 프로젝트 이름 : 1, 2, 3, 4, 5, 6
- 실행화면을 캡처하여 한글파일에 추가 후 솔루션 폴더에 포함.
- 한글 파일명 : 학번_이름.hwp
- 솔루션 폴더를 압축하여 제출할 것.
- 솔루션 압축 파일 명:
 - 1차 제출: 학번_이름_DS_03(1).zip, 2차 제출: 학번_이름_DS_03(2).zip
- 제출은 2회 걸쳐 가능(수정 시간 기준으로 처리)