

## 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 100 보다 작은 값으로 메모리를 초기화 시킨 후, 주소값을 반환한다. (즉 함수 호출로 1차원 배열 만들기)

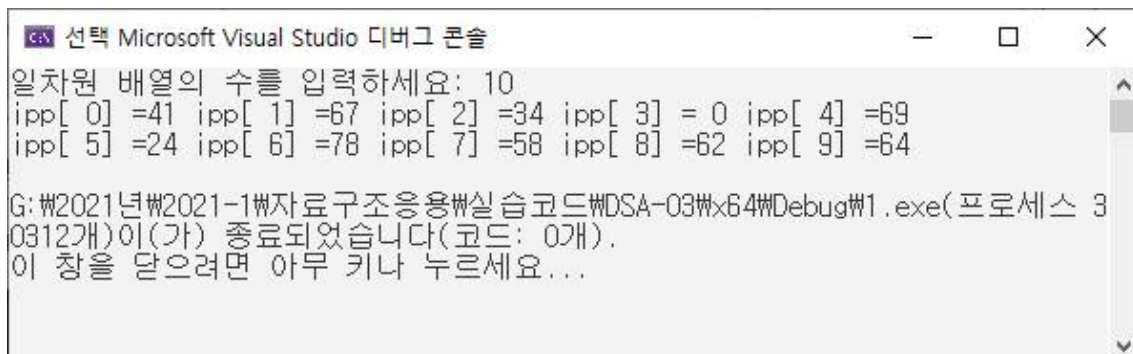
```
int main()
{
    int *ipp;
    int i, num;

    printf("일차원 배열의 수를 입력하세요: ");
    scanf_s("%d", &num);2

    ipp = make1dArray(num);
    for (i = 0; i < num; i++) {10 10
        printf("ipp[%2d] =%2d ", i, ipp[i]);
        if ((i % 5) == 4)
            printf("\n");

    }
    return 0;
}
```

#### [실행]

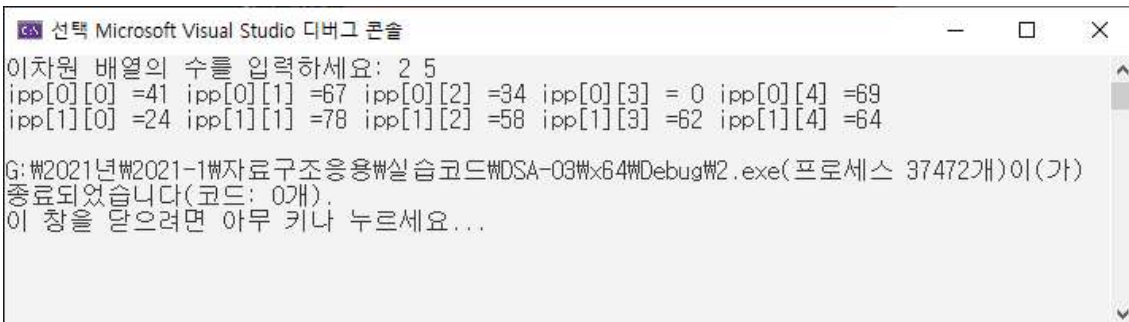


```
선택 Microsoft Visual Studio 디버그 콘솔
일차원 배열의 수를 입력하세요: 10
ipp[ 0] =41 ipp[ 1] =67 ipp[ 2] =34 ipp[ 3] = 0 ipp[ 4] =69
ipp[ 5] =24 ipp[ 6] =78 ipp[ 7] =58 ipp[ 8] =62 ipp[ 9] =64
G:\2021년\2021-1\자료구조응용\실습코드\DSA-03\64\Debug\1.exe(프로세스 30912개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

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 랜덤으로 100보다 적은 값으로 메모리를 초기화 시킨 후, 주소값을 반환한다.

```
int main()
{
    int **ipp;
    int i, j, rows, cols;
    printf("이차원 배열의 행과 열의 수를 입력 하세요 : ");
    scanf_s("%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
}
```

#### [실행]



```
선택 Microsoft Visual Studio 디버깅 콘솔
이차원 배열의 수를 입력하세요: 2 5
ipp[0][0] =41 ipp[0][1] =67 ipp[0][2] =34 ipp[0][3] = 0 ipp[0][4] =69
ipp[1][0] =24 ipp[1][1] =78 ipp[1][2] =58 ipp[1][3] =62 ipp[1][4] =64
G:\2021년\2021-1\자료구조응용\실습코드\DSA-03\64\Debug\2.exe(프로세스 37472개)이(가)
종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

3. [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;
}

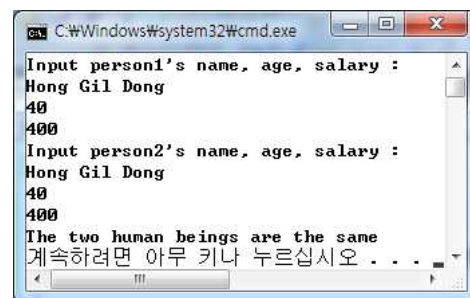
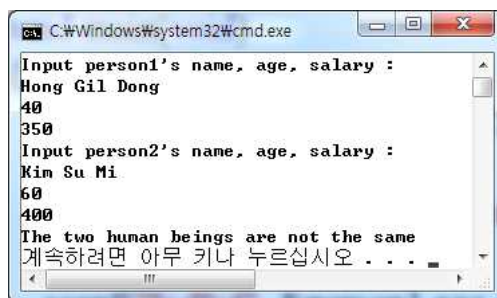


---


```

Program 2.4: Function to check equality of structures

[실행]



※ 공백을 포함한 이름을 입력하여야 함

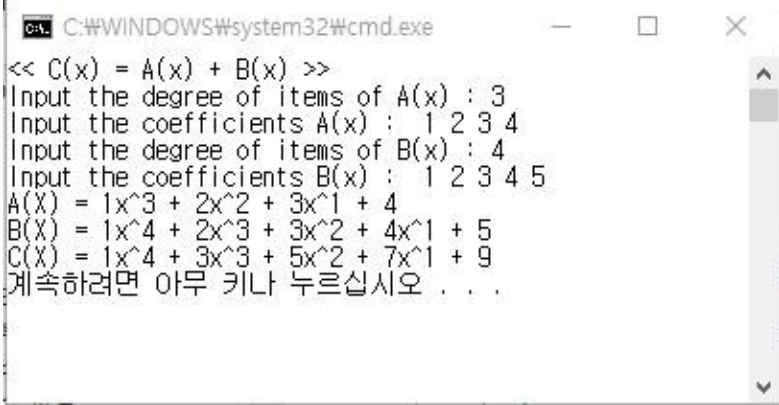
4. 다항식을 아래와 같이 표현 하는 방식을 이용하여, 두 다항식을 입력받아 더한 후 그 결과를 출력하는 프로그램을 작성하라.

```
#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);`

#### [실행]



```
C:\WINDOWS\system32\cmd.exe
<< C(x) = A(x) + B(x) >>
Input the degree of items of A(x) : 3
Input the coefficients A(x) : 1 2 3 4
Input the degree of items of B(x) : 4
Input the coefficients B(x) : 1 2 3 4 5
A(X) = 1x^3 + 2x^2 + 3x^1 + 4
B(X) = 1x^4 + 2x^3 + 3x^2 + 4x^1 + 5
C(X) = 1x^4 + 3x^3 + 5x^2 + 7x^1 + 9
계속하려면 아무 키나 누르십시오 . . .
```

5. 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`, `attatch` 함수 참조
- ③ 다항식을 출력하는 함수를 정의하여 사용하기

[실행]

```
<< D(x) = A(x) + B(x) >>
Input the number of items of A(x) : 2
Input the number of items of B(x) : 3

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

A(x) = 10x^10 + 5x^5
B(x) = 10x^10 + 5x^5 + 7
D(x) = 20x^10 + 10x^5 + 7

G:\2022년\2022-1\자료구조응용\실습코드\DSA-03\64\Debug\5.exe(프로세스 20160개)이(가)
종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

#### ■ 제출 형식

- 솔루션 이름 : 학번\_이름\_DS\_03
- 프로젝트 이름 : 1, 2, 3, 4, 5
- 솔루션 폴더를 압축하여 제출할 것.
- 학습관리시스템에 과제를 올릴 때 제목:
  - 1차 제출: 학번\_이름\_DS\_03(1), 2차 제출: 학번\_이름\_DS\_03(2)
  - 제출은 2회걸쳐 가능(수정 시간 기준으로 처리)