

함수 관련 문제 #1

1. 연도를 입력받아 해당 년도가 윤년인지 아닌지 판단하는 프로그램을 작성하시오.

- 입력받은 연도 year
- 윤년 여부를 판단하는 IsLeapYear 함수 정의
- IsLeapYear 함수는 윤년이면 1을, 그렇지 않으면 0을 리턴

(연도 를 4로 나누어 떨어지면 윤년, 4로 나누어 떨어지는 연도 중에서 100으로 나누어 떨어지는 연도는 윤년이 아니고, 100으로 나누어 떨어지는 연도 중 다시 400으로 나누어 떨어지는 연도는 윤년)

코드

```
#include<stdio.h>
int IsLeapYear(int n); // 함수 선언
int main()
{
    int year;

    printf("연도를 입력하세요 : ");
    scanf_s("%d", &year); // 연도 입력

    if (IsLeapYear(year) == 1) {
        printf("윤년입니다.\n");
    } // 함수 호출 후 1이 반환되면 윤년임을 알려줌
    else {
        printf("윤년이 아닙니다.\n");
    } // 함수 호출 후 0이 반환되면 윤년이 아님을 알려줌

    return 0;
}

int IsLeapYear(int n) // 함수 정의
{
    if (n % 4 == 0 && n % 100 != 0 || n % 400 == 0) {
        return 1; // 전달받은 연도가 윤년 공식에 성립하면 1반환
    }
    else {
        return 0; // 성립하지 않으면 0반환
    }
}
```

실행 결과

 Microsoft Visual Studio 디버그 콘솔

```
연도를 입력하세요 : 2020
윤년입니다.
```

Microsoft Visual Studio 디버그 콘솔

연도를 입력하세요 : 2019
윤년이 아닙니다.

함수 관련 문제 #2

2. 5개의 정수를 입력받아 그 중에서 가장 큰 수를 출력하는 프로그램을 작성하시오.

- 5개의 정수를 입력받는 배열 arr
- 전달 받은 수 중 가장 큰 값을 반환하는 PrintMax 함수 정의

코드

```
#include<stdio.h>
int PrintMax(int* arr, int n) // 함수 정의
{
    int max = 0; // 가장 큰 수를 0으로 초기화
    for (int i = 0; i < n; i++)
    {
        if (arr[i] > max)
            max = arr[i];
    } // 전달받은 배열의 원소가 max보다 크면 max에 저장
    return max; // max 반환
}
int main()
{
    int arr[5];
    printf("5개의 정수를 입력하시오 : ");
    for (int i = 0; i < 5; i++)
    {
        scanf_s("%d", &arr[i]); // 배열에 5개의 정수 입력
    }
    int result = PrintMax(arr, 5);
    // Sum 함수를 호출하여 반환되는 큰 수를 result에 저장

    printf("입력되어 있는 수에서 가장 큰 수는 %d 입니다.\n", result);
    return 0;
}
```

출력 결과

Microsoft Visual Studio 디버그 콘솔

5개의 정수를 입력하시오 : 11 23 45 34 31
입력되어 있는 수에서 가장 큰 수는 45 입니다.

함수 관련 문제 #3

3. 두 개의 문자열을 입력받아 비교하는 프로그램을 작성하시오.

- 두 개의 문자열을 입력받는 배열 arr1, arr2 (대문자X)
- 두 문자열을 전달받아 비교하는 Compare 함수 정의
- Compare 함수는 앞의 문자열이 더 크면 1, 뒤의 문자열이 더 크면 -1 반환
- 반환 된 값에 따라 arr1과 arr2 문자열 비교 (출력)

코드

```
#include <stdio.h>

int Compare(char* arr1, char* arr2); // 함수 선언

int main()
{
    char arr1[50], arr2[50];
    printf("첫번째 문자열을 입력하세요: ");
    gets(arr1); // arr1에 첫번째 문자열 입력
    printf("두번째 문자열을 입력하세요: ");
    gets(arr2); // arr2에 두번째 문자열 입력

    if (Compare(arr1, arr2) == 1) {
        printf("%s이(가) %s보다 뒤에 있습니다.\n", arr1, arr2); // Compare 함수
        호출 후 1이 반환되면 arr1이 arr2보다 뒤에 있다고 출력
    }
    else {
        printf("%s이(가) %s보다 앞에 있습니다.\n", arr1, arr2); // Compare 함수
        호출 후 -1이 반환되면 arr1이 arr2보다 앞에 있다고 출력
    }
    return 0;
}

int Compare(char* arr1, char* arr2) // 함수 정의
{
    while (*arr1 != '\0' || *arr2 != '\0') // arr1과 arr2 중 짧은 문자열의 길이만큼
    반복
    {
        if (*arr1 > *arr2) {
            return 1; // arr1의 문자가 arr2의 문자보다 크면 1반환
        }
        else if (*arr1 < *arr2) {
            return -1; // arr1의 문자가 arr2의 문자보다 작으면 -1반환
        }
        *arr1++;
        *arr2++; // 비교하는 arr1과 arr2의 문자가 같으면 다음 문자로 넘어감
    }
}
```

실행 결과

```
Microsoft Visual Studio 디버그 콘솔
첫번째 문자열을 입력하세요: teacher
두번째 문자열을 입력하세요: student
teacher이(가) student보다 뒤에 있습니다.
```

```
Microsoft Visual Studio 디버그 콘솔
첫번째 문자열을 입력하세요: student
두번째 문자열을 입력하세요: teacher
student이(가) teacher보다 앞에 있습니다.
```

배열 관련 문제 #1

1. 5개의 정수를 입력받아 n만큼 증가시킨 후 결과를 나타내는 프로그램을 작성하시오.

- 5개의 정수를 입력받는 배열 arr
- 입력받은 증가량 n
- arr의 원소에 각각 n씩 증가시킨 결과 출력

코드

```
#include<stdio.h>
int main()
{
    int arr[5];
    int n;

    printf("5개의 정수를 입력하세요 : ");
    for (int i = 0; i < 5; i++) {
        scanf_s("%d", &arr[i]); // 배열에 5개의 정수 입력
    }

    printf("증가량을 입력하세요 : ");
    scanf_s("%d", &n); // 배열의 각 원소에 더해줄 증가량 n 입력

    printf("증가한 후의 정수 : ");
    for (int i = 0; i < 5; i++) {
        printf("%d ", arr[i] += n); // 배열의 각 원소에 n씩 더해서 출력
    }

    return 0;
}
```

출력 결과

Microsoft Visual Studio 디버그 콘솔

```
5개의 정수를 입력하세요 : 10 20 30 40 50
증가량을 입력하세요 : 5
증가한 후의 정수 : 15 25 35 45 55
```

배열 관련 문제 #2

2. 10개의 정수 중 중복이 한번도 되지 않는 수를 구하는 프로그램을 작성하시오.

- 10개의 정수를 입력 받는 배열 arr
- 이중 for문을 사용하여 자신과 중복되는 수를 발견하면 count++ (자기 자신과도 비교)
- 중복되는 수가 없는 수 출력

코드

```
#include<stdio.h>
int main()
{
    int arr[10];
    printf("10개의 정수를 입력하세요 : ");
    for (int i = 0; i < 10; i++) {
        scanf_s("%d", &arr[i]); // arr에 10개의 정수 입력
    }
    printf("중복되지 않은 수 : ");
    for (int i = 0; i < 10; i++){
        int count = 0; // 매 반복마다 count를 0으로 초기화
        for (int j = 0; j < 10; j++){
            if (arr[i] == arr[j]) {
                count++; // 배열의 원소와 자신포함 다른 원소와 중복이
                있으면 count+1
            }
        }
        if (count == 1) {
            printf("%d ", arr[i]); // count가 1일 경우, 다른 원소와 중복이
            되지 않는 것이므로 출력
        }
    }
    return 0;
}
```

출력 결과

Microsoft Visual Studio 디버그 콘솔

```
10개의 정수를 입력하세요 : 10 10 40 20 10 80 50 80 40 30
중복되지 않은 수 : 20 50 30
```

배열 관련 문제 #3

3. 5개의 문자열 중 가장 긴 문자열과 그 길이를 출력하는 프로그램을 작성하시오.

- 5개의 문자열을 입력받는 이차원배열 str
- 이중 for문을 이용하여 각각의 문자열 길이 측정
- 문자열 길이 중 가장 큰 값 max
- 가장 긴 문자열의 위치 index
- index에 위치한 문자열과 그 길이 출력


코드

```
#include<stdio.h>
int main()
{
    char str[5][80];
    int max = 0, index = 0;
    printf("5개의 문자열을 입력하시오\n");
    for (int i = 0; i < 5; i++)
    {
        gets(str[i]); // 이차원 배열 str에 5개의 문자열 입력
    }
    for (int i = 0; i < 5; i++)
    {
        int count = 0; // 매 반복마다 count는 0으로 초기화
        for (int j = 0; str[i][j]; j++)
        {
            count++; // i에 위치한 문자열의 길이 count
        }
        if (max < count)
        {
            max = count; // max보다 count가 크면 count를 max에 저장
            index = i; // index에 가장 긴 문자열의 위치 저장
        }
    }
    printf("가장 긴 문자열 : ");
    puts(str[index]); // 가장 긴 문자열 출력
    printf("문자열의 길이는 %d 입니다.\n", max); // 가장 긴 문자열의 길이 출력
    return 0;
}
```

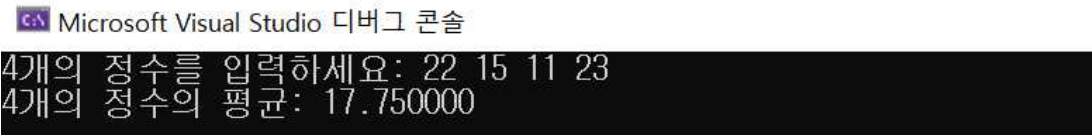
실행 결과

```
Microsoft Visual Studio 디버그 콘솔
5개의 문자열을 입력하시오
Hi
My name is JiYun.
Dankook is the best!
My favorite season is Winter.
C programming is fun!
가장 긴 문자열 : My favorite season is Winter.
문자열의 길이는 29 입니다.
```

포인터 관련 문제 #1
1. 포인터 변수를 이용하여 10개의 입력받은 정수 중 짝수와 홀수의 개수를 구하는 프로그램을 작성하시오. - 10개의 정수를 입력받는 배열 a와 a의 주소를 저장하는 포인터변수 p - p의 간접참조값이 홀수인지 짝수인지 판단하여 각각 카운트 - 홀수의 개수와 짝수의 개수 출력
코드
<pre>#include<stdio.h> int main() { int a[10]; int *p = a; // 포인터 배열 p에 a의 주소 저장 int odd = 0; int even = 0; printf("10개의 정수 입력: "); for (int i = 0; i < 10; i++) { scanf_s("%d", &a[i]); // a에 10개의 정수 입력 } for (int i = 0; i < 10; i++) { if ((*p) % 2 == 0) { even++; // p의 간접 참조 값이 짝수이면 even +1 } else { odd++; // p의 간접 참조 값이 홀수이면 odd +1 } } }</pre>

<pre> *p++; // 다음 원소로 접근 } printf("짝수 : %d개\n", even); // 짝수 개수 even 출력 printf("홀수 : %d개\n", odd); // 홀수 개수 odd 출력 return 0; } </pre>
실행 결과
 <pre> Microsoft Visual Studio 디버그 콘솔 10개의 정수 입력: 9 2 5 2 1 6 8 10 7 3 짝수 : 5개 홀수 : 5개 </pre>

포인터 관련 문제 #2
<p>2. 포인터 변수를 이용하여 4개의 정수의 평균을 구하는 프로그램을 작성하시오.</p> <ul style="list-style-type: none"> - 4개의 정수를 입력받는 배열 a와 a의 주소를 저장하는 포인터 변수 p - 4개의 정수의 합을 저장하는 변수 sum - 4개의 정수의 평균을 저장하는 변수 avg
코드
<pre> #include<stdio.h> int main() { int a[4]; int* p = a; // 포인터 변수 p 에 a의 주소 저장 int n, sum = 0; printf("4개의 정수를 입력하세요: "); for (int i = 0; i < 4; i++) { scanf_s("%d", &a[i]); // 배열 a에 4개의 정수 입력 } for (int i = 0; i < 4; i++) { sum = sum + *p; // sum에 p의 간접참조값(a의 원소) 더해줌 *p++; // 다음 원소로 접근 } double avg = sum / 4.0; // 실수avg 에 평균 저장 </pre>

<pre>printf("4개의 정수의 평균: %lf\n", avg); // 평균 출력 return 0; }</pre>
실행 결과
 <p>Microsoft Visual Studio 디버그 콘솔</p> <pre>4개의 정수를 입력하세요: 22 15 11 23 4개의 정수의 평균: 17.750000</pre>

포인터 관련 문제 #3
<p>3. 포인터 변수를 이용하여 입력받은 문자열 중 특정 문자가 몇 개 있는지 출력하는 프로그램을 작성하시오.</p> <ul style="list-style-type: none"> - 문자열을 입력받는 배열 str과 str의 주소를 저장하는 포인터 변수 find - 찾고 싶은 문자를 입력받는 변수 n - 반복문을 통해 문자열에 해당 문자가 몇 개인지 count
코드
<pre>#include<stdio.h> int main() { char str[80]; char* find; find = &str; // 포인터 변수 초기화 char n; int count = 0; printf("문자열을 입력하시오 : "); gets(str); // 배열 str에 문자열 입력 printf("찾고 싶은 문자는? "); scanf_s("%c", &n); // 찾고싶은 문자 n 입력 for (int i = 0; str[i]; i++) { if (*(find + i) == n) count++; // i번째 문자가 n과 일치하면 count+1 } printf("찾으려는 문자의 개수는 %d개입니다.\n", count); // 찾으려는 문자 개수 출력 return 0; }</pre>
실행 결과

```
Microsoft Visual Studio 디버그 콘솔
문자열을 입력하시오 : show me the money
찾고 싶은 문자는? e
찾으려는 문자의 개수는 3개입니다.
```

문자열처리 관련 문제 #1

1. 문자열을 역순으로 출력하는 프로그램을 작성하시오. (문자열 처리 라이브러리 함수 이용)
- 문자열을 입력받는 배열 str
 - 문자열 길이를 구하는 라이브러리 함수 strlen()를 이용하여 문자열 역순 출력

코드

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[80];
    printf("문자열을 입력하시오 : ");
    gets(str); // 배열 str에 문자열 입력

    int length = strlen(str) - 1; // length에 널값을 제외한 문자열의 길이 저장

    printf("문자열의 역순 출력 : ");
    for (int i = length; i >= 0; i--){
        printf("%c", str[i]); // str을 인덱스 length 부터 0까지 역순으로 출력
    }
    return 0;
}
```

실행 결과

```
Microsoft Visual Studio 디버그 콘솔
문자열을 입력하시오 : ABCDEFGHIJKLMNOPQRSTUVWXYZ
문자열의 역순 출력 : ZYXWVUTSRQ PONMLKJIHGFEDCBA
```

문자열처리 관련 문제 #2

2. 문자열을 입력받아 대문자는 소문자로, 소문자는 대문자로 변환하는 프로그램을 작성하시오. (문자열 처리 라이브러리함수 이용)

- 문자열 입력받는 배열 str
- isupper() 와 islower()를 이용하여 대문자인지 소문자인지 판단
- 대문자는 tolower(), 소문자는 toupper()를 이용하여 변환

코드

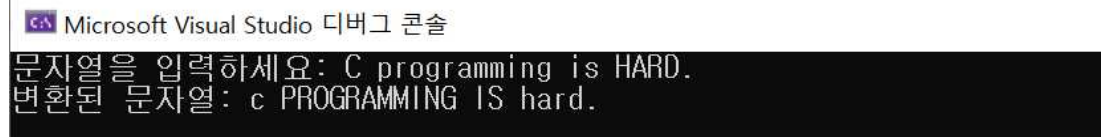
```
#include <stdio.h>
#include <string.h>

int main()
{
    char str[80];
    printf("문자열을 입력하세요: ");
    gets(str); // str에 문자열 입력

    for (int i = 0; str[i]; i++) {
        if (isupper(str[i])) {
            str[i] = tolower(str[i]); // str의 i번째 문자가 대문자이면 소문자
            로 변환하여 저장
        }
        else if (islower(str[i])) {
            str[i] = toupper(str[i]); // str의 i번째 문자가 소문자이면 대문자
            로 변환하여 저장
        }
    } // 문자열이 끝날때까지 반복

    printf("변환된 문자열: ");
    puts(str); // 변환 후 문자열 출력
}
```

실행 결과



Microsoft Visual Studio 디버그 콘솔

문자열을 입력하세요: C programming is HARD.
변환된 문자열: c PROGRAMMING IS hard.

문자열처리 관련 문제 #3

3. 5개의 단어 중 가장 큰 단어와 가장 작은 단어를 병합하는 프로그램을 작성하시오. (문자열 처리 라이브러리 함수 이용)

- 5개의 단어를 입력받는 이차원 배열 arr
- 가장 큰 단어의 인덱스를 저장하는 indexmax, 가장 작은 단어의 인덱스를 저장하는 indexmin
- strcmp(,)를 이용하여 단어 비교
- strcat(,)를 이용하여 단어 붙이기

코드

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

int main()
{
    char arr[5][20];
    char result[40];
    int indexmax=0; // 가장 큰 문자 인덱스 0으로 초기화
    int indexmin=0; // 가장 작은 문자 인덱스 0으로 초기화

    for (int i = 0; i < 5; i++) {
        printf("%d번째 단어 입력: ",i+1);
        gets(arr[i]); // 이차원 배열 arr에 5개의 단어 입력
    }
    for (int i = 1; i < 5; i++) {
        if (strcmp(arr[indexmax], arr[i])<0) {
            indexmax = i; // indexmax번째 단어보다 i번째 단어가 크면 가장 큰 문자 인덱스는 i
        }
        if (strcmp(arr[indexmax], arr[i]) > 0) {
            indexmin = i; // indexmin번째 단어보다 i번째 단어가 작으면 가장 작은 문자 인덱스는 i
        }
    }

    printf("%s + %s 결과 : ",arr[indexmax],arr[indexmin]);
    strcat(arr[indexmax], arr[indexmin]); // 가장 큰 단어에 가장 작은 단어 붙여넣기
    puts(arr[indexmax]); // 병합 결과 출력
    return 0;
}
```

실행 결과

Microsoft Visual Studio 디버그 콘솔

```
1번째 단어 입력: hello  
2번째 단어 입력: world  
3번째 단어 입력: cute  
4번째 단어 입력: dog  
5번째 단어 입력: jiyun  
world + cute 결과 : worldcute
```