

프논이 2020 학기말 고사

■ 문제 1

정수 n (1이상 20이하)을 입력받고
정수형배열의 원소를 n 개 입력받고

key 값을 입력받은후

이 배열안에 key 값이 몇개 있는 가를 출력하는 프로그램을 작성하라.

실행예:

입력)

6 <- 원소의 개수

11 22 33 44 33 66 <- 6개의 원소

33 <- key값

출력)

2

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

```
int main(void)
```

```
{
```

```
    int a[20];
```

```
    int n, key;
```

```
    int i;
```

```
    scanf("%d", &n);
```

```
    for (i = 0; i < n; i++)
```

```
        scanf("%d", &a[i]);
```

```
    scanf("%d", &key);
```

```
    //여기 코드를 완성하시오.
```

```
}
```

■ 문제 2

문제(소수 비소수 구분)(배열)(난이도 중)

정수 n (20 이하)을 입력으로 받고,

정수형 배열의 원소를 1 보다 큰 수들로 n 개 입력받아,

이들 정수를 소수와 소수가 아닌 것으로 구분하려한다.

소수(들)을 먼저 출력한 후에 소수가 아닌 것(들)을 출력하는 프로그램을 작성하라.

[주의]

- 변경하지말라 등 지시사항이 없는 경우 코드를 수정해도 좋음
- 변수 추가 가능
- 소수와 비소수를 같은 줄에 출력하고 수와 수 사이에 공백 1개를 둔다.

[힌트]

is_prime 함수를 정의해 사용하면 편리하다.(안해도 됨)

실행예:

입력)

6 <- 배열의 크기

11 12 13 14 15 16 <- 배열의 원소

출력)

11 13 12 14 15 16 <- 소수인 11, 13을 출력하고 소수 아닌 정수를 출력

// num이 소수이면 1을 아니면 0을 반환하는 함수를 사용하면 편리하다.

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

```
int is_prime(int num)
```

```
{
```

```
}
```

```
int main(void)
```

```
{
```

```
    int a[20], p[20], np[20]; // p는 소수를 np는 소수아닌 수를 담을 배열
```

```
    int i;
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    for (i = 0; i < n; i++)
```

```
        scanf("%d", &a[i]);
```

```
}
```

■ 문제 3

문제(자리수중홀수합)(난이도 중)

정수 n을 읽어서
이를 3배 한 후
각 자리 수들 중 홀수만 더하여 합을 출력하는 프로그램을 작성하라.
main은 변경하지말고 cal_odd_digit만 완성하라.(변경시 감점 처리)

[주의]

- 변경하지말라 등 지시사항이 없는 경우 코드를 수정해도 좋음
- 변수 추가 가능

실행예 1:

입력)

123

출력)

12 (=3 + 9: 123 * 3 = 369 이고 이중 3과 9가 홀수이므로)

실행예 2:

입력)

1451

출력)

11 (= 3 + 5 + 3: 1451 * 3 = 4353 이고 이중 3 5 3이 홀수이므로)

```
#include <stdio.h>
int cal_odd_digit(int num)
{
}
```

```
int main(void) // main은 변경하지 않는다. 변경시 감점 처리
{
    int n;

    scanf("%d", &n);
    n *= 3;

    printf("%d\n", cal_odd_digit(n));
}
```

■ 문제 4

문제(word제외알파벳나열)(난이도 중)

알파벳 단어는 A부터 Z이다.

단어(길이가 20 미만인)를 읽어 들인 후
알파벳(A부터 Z까지) 중 그 단어에 속하지 않은 문자들을 차례대로 나열하는 프로그램을 작성하라.

즉, BOY를 단어로 입력받으면, B, O, Y를 제외한 나머지 알파벳을 다음과 같이 공백없이 나열한다.
ACDEFGHIJKLMNOPQRSTUVWXYZ

[주의]

- 변경하지말라 등 지시사항이 없는 경우 코드를 수정해도 좋음
- 변수 추가 가능
-

[힌트]

문자열 w 안에 ch가 있으면 1을 없으면 0을 반환하는 함수 is_in()함수를 정의하여 main()함수에서
호출하여 사용한다.
(is_in()함수 사용 안해도 됨)

실행예

입력)

BOY

출력)

ACDEFGHIJKLMNOPQRSTUVWXYZ

```
#include <stdio.h>
// ch가 w안에 있으면 1을, 없으면 0을 반환한다
int is_in(char w[], char ch) // 사용 안해도 됨
{
}
```

```
int main(void)
{
    char word[20];

    scanf("%s", word);
```

```
}
```

■ 문제 5

문제(느슨한 패스워드 체크)

프논이 교수는 apple이라는 패스워드를 쓴다.

로그인시 패스워드를 입력할 때 자꾸 apple뒤에 뭔가를 붙이는 이상한 습관이 생겼다.
패스워드 입력시 앞부분만 맞으면 로그인이 성공 하도록 <느슨한 패스워드 체크 시스템>으로 바꾸려한다.

즉 apple을 넣으면 당연히 로그인 성공이지만 apple1, 혹은 appleapple이라고 넣어도 로그인 성공이 되게 하고 싶다.

이러한 느슨한패스워드체크를 테스트하기 위해 패스워드를 3번 입력하게 하고 각 패스워드에 대해서 성공하면 1을 실패하면 0을 출력한다.

예를 들어, 세 번 모두 맞는 패스워드이면 111을,
처음 시도만 맞으면 100, 다 틀리면 000을 출력한다.

[주의]

- 변경하지말라 등 지시사항이 없는 경우 코드를 수정해도 좋음
- 변수 추가 가능

실행예 1:
입력) apple apple appleb <- 세 개의 입력 모두 패스워드로 성공
출력) 111

실행예 2:
입력) apple apple bple <- 첫번째 시도만 패스워드 입력 성공
출력) 100

```
#include <stdio.h>
int checking(char w0[], char w1[])
{
}
int main(void)
{
    char w0[81] = "apple";
    char w1[81], w2[81], w3[81];

    scanf("%s %s %s", w1, w2, w3);

    printf("%d", checking(w0, w1));
    printf("%d", checking(w0, w2));
    printf("%d", checking(w0, w3));
}
```

■ 문제 6

문제(단어속단어여부)(난이도 상)

is_word_in_word 함수는 다음과 같은 prototype을 갖는다.

```
int is_word_in_word(char w1[], int start, char w2[]);
```

위의 함수는 단어 w2가 단어 w1의 start 인덱스위치에 있으면 1을, 아니면 0을 반환한다.

예를 들어 w1가 ababcd이고 w2가 abc일 때,
start가 2이면 위의 함수는 1을 반환하고
그 외에는 0을 반환한다.

주어진 프로그램의 main 함수는

문자열 s1과 s2를 입력으로 받아서
start를 0부터 (w1의 길이 - 1)까지 반복적으로 is_word_in_word를 호출하여 그 반환값(0 또는 1)을
출력한다.

즉 s1가 ababcd, s2가 abc라면

```
start가 0일때: 0
start가 1일때: 0
start가 2일때: 1 <- s1에서 인덱스가 2인 위치에 s2가 존재하므로
start가 3일때: 0
start가 4일때: 0
start가 5일때: 0
```

그러므로 001000을 출력한다.Copy

is_word_in_word 함수를 정의하여 아래와 같은 실행예가 되도록 프로그램하라.

[주의]

- main 함수는 그대로두고(변경시 감점)
- 변경하지말라 등 지시사항이 없는 경우 코드를 수정해도 좋음
- 변수 추가 가능

실행예
입력) ababcd abc
출력) 001000

실행예
입력) cabbcc cc
출력) 000010

실행예
입력) abcabc abc
출력) 100100

실행예
입력) abcdef xx
출력) 000000

```
#include <stdio.h>

// 문자열 w2가 문자열 w1의 start 위치에 있으면 1을, 아니면 0을 반환하는 함수

int is_word_in_word(char w1[], int start, char w2[])
{

}

int main(void) // main은 변경하지 말고 사용한다. 변경시 감점
{
    char s1[20], s2[20];

    int i, j;

    scanf("%s", s1);
    scanf("%s", s2);

    for (i = 0; s1[i] != '\0'; i++)
        printf("%d", is_word_in_word(s1, i, s2));
}
```