

5문제(100점) + bonus문제(50점)

1. (10점)소수의 합(감시값반복문)(난이도 하): 코드 전체를 제출

-1 이 나올 때까지 정수(2 이상의))를 반복적으로 읽어 정수가 소수인 경우 합을 계산하려 한다.

실행예 1
2 3 4 5 6 7 8 9 -1 <- 입력
17 <- 출력: 2 + 3 + 5 + 7

실행예 2
11 22 33 -1 <- 입력
11 <- 출력: 11 만 소수

실행예 3
-1 <- 입력
0 <- 출력

2. (10 점)단어 속 문자 위치(문자열)(난이도 하) : 코드 전체를 제출

문자열(크기가 20 미만)을 입력 받고
key문자를 입력 받은 후 key문자가
이 문자열 안에 몇 번째에 있는지를 출력하는 프로그램을 작성하라.
key문자가 문자열 안에 없는 경우 -1을 출력한다.
key가 여러 번 나오는 경우 가장 앞에 것에 대한 위치를 출력한다.

실행예1
abcdefg c <- 입력: 문자열과 key문자
3 <- 출력: c는 3번째에 있음

실행예2
abcdefg x <- 입력: 문자열과 key문자
-1 <- 출력: x는 없음

힌트:
문자열을 입력받기 위해 아래와 같이 선언하고 입력 받는다.
char s[20];
char key;
scanf("%s %c", s, &key);

3. (20점)상위하위 분류(배열)(난이도 중) : 코드 전체를 제출

배열의 개수 n(≤ 20)을 입력하고, 배열의 개수만큼 점수(들)를 입력받은 후
기준점수를 하나 입력받아
그 기준점수 미만의 점수들을 먼저 출력한 후
기준점수 이상의 점수들을 출력하는 프로그램을 작성하라.
점수들을 출력할 때 입력 받은 순서를 유지한다.

실행예:
입력)
10 <- 배열의 개수
2 9 3 8 4 7 5 6 1 10 <- 배열의 원소
5 <- 기준점수
출력)
2 3 4 1
9 8 7 5 6 10

- 출력할 때 숫자 사이에 공백을 1개씩 둔다
- 기준점수 미만의 점수들을 출력한 후에 줄을 바꾼 후
기준점수 이상의 점수들을 출력한다

4. 숫자로 암호화

대문자들에 대해서
A는 1, B는 2, C는 3, ..., Z는 26으로 수를 부여할 수 있다.

주어진 단어(대문자로 구성된) (크기가 20 미만)에 대해서 위의 매칭 방법으로 암호화하려한다.
예를들어 주어진 단어가 ABD이면 암호화 결과는 124가 된다.

가정1. 단어는 대문자로만 이루어진다.

4-1. (20점)A부터 I까지 암호화(난이도 중) : 함수만 제출

아래의 가정을 추가하여 프로그램을 작성하라.

가정2. A(1로 매칭)부터 I(9로 매칭)까지의 문자로만 이루어진다.

대문자로 이루어진 문자열 s를 위의 암호화규칙을 적용하여 숫자들로 된 문자열 e로 변환하는 함수
changeToDigitEncode를 작성하여 이를 제출하라. 원형은 아래와 같다.
void changeToDigitEncode(char s[], char e[]);

실행예1
ABBA<- 입력
1221 <- 출력: 숫자로 암호화(A:1, B:2)

실행예2
BIG<- 입력
297 <- 출력: 숫자로 암호화(B:2, I:9, G:7)

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

void changeToDigitEncode(char s[], char e[]) // 이 함수를 제출
{
    // 코드 작성
}

int main(void) // 변경하지 마라
{
    char sentence[20], encoded[40];

    scanf("%s", sentence);
    changeToDigitEncode(sentence, encoded);
    printf("%s\n", encoded);
}
```

4-2.(15점) A부터 Z까지 암호화(난이도 중상): 함수만 제출

위의 4-1에서의 가정 2를 아래와 같이 바꾸어 프로그램을 작성하라.

수정된 가정2. 단어는 A부터 Z의 문자로 이루어진다

A-Z의 문자로 이루어진 문자열 s를 숫자들로 된 문자열 e로 바꾸는
함수 changeToDigitEncode를 작성하여 이를 제출한다.
4-1의 뼈대코드를 그대로 이용하라.

```
실행예1
AZ<- 입력
126 <- 출력: 숫자로 암호화(A:1, Z:26)
```

```
실행예2
BTS <- 입력
22019 <- 출력: 숫자로 암호화(B:2, T:20, S:19)
```

5. (30점)숨은 암호 찾기(문자열)(난이도 중상): 함수만 제출

크기가 20 미만인 문자열(input)에 크기가 20 미만인 암호(passwd)가 숨어있는 가를 판별하는 함수
isHiddenPasswd 를 작성하려 한다.
passwd 가 input 안에 순서대로 나타나면 1 을 아니면 0 을 반환한다.

이때, input 에는 다른 문자들이 섞여있어도 된다.
예를 들어서 passwd 가 apple 일 때 input 이 abpple 이면 암호가 숨겨진 것으로 간주한다.
또한 input 이 bappxe 이거나 capple 일 때도 역시 암호가 숨겨진 것으로 간주한다.
반면에 appld, appel 은 숨겨진 암호가 아니다.

암호를 읽어 들인 후
입력되는 3 개의 input 에 대해서 숨겨진 암호 여부에 따라 0(아닌경우) 혹은 1(맞는 경우)를 출력한다.
주어진 main 을 변경하지 말고 아래와 같은 실행예를 갖도록 함수 isHiddenPasswd 를 작성하여 이를 제출한다.

```
실행예 1
apple <- 입력: 암호
abpple apple appld <- 입력: 3 개의 input
110 <- abpple, apple 은 숨겨진 암호이고 appld 는 아니다
```

```
실행예 2
banana <- 입력: 암호
baanaana appel abcde <- 입력: 3 개의 input
100 <- baanaana 은 숨겨진암호이고 appel, abcde 는 아니다
```

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int isHiddenPassword(char passwd[], char input[]) // 이 함수를 제출
{
    // 코드 작성
}

int main(void) //변경하지 마라
{
    char passwd[20], input[20];
    int i;

    scanf("%s", passwd);
    for (i = 0; i < 3; i++) {
        scanf("%s", input);
        if (isHiddenPassword(passwd, input))
            printf("1");
        else
            printf("0");
    }
    printf("\n");
}
```