

Jaewon Choi  
Keunsan Park  
Haeun Lee  
(snucsl.ta@gmail.com)

Systems Software &  
Architecture Lab.

Seoul National University

Spring 2023

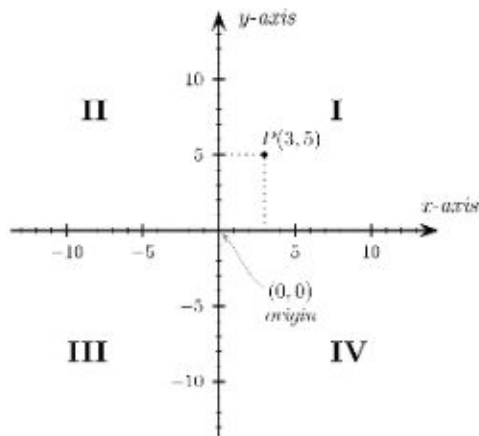
# 4190.103A-001: Programming Practice Lab. 5



# Lab. 4 실습 풀이

# 실습 1

Q. 임의의 두 정수  $x, y$ 가 주어졌을 때 해당  $(x,y)$  좌표가 몇 사분면에 속하는지 출력해보자.



사분면은 위 그림처럼 1 ~ 4의 숫자로 표현 가능하다.

임의의 두 정수가 한 줄로 주어질 때, 띄어쓰기를 기준으로 앞의 숫자가  $x$ , 뒤의 숫자가  $y$ 라고 하자. 해당 좌표가 몇사분면에 속하는지

`#define` 으로 정의된 변수와 논리 연산자를 활용하여 출력해보자.

(단, 원점 및  $x$ - $y$ 축 위에 있을 경우 '-'을 출력하고, `TWO, THREE, FOUR` 를 출력할 땐 `%s` 를 사용하자.)

```
1 #include<stdio.h>
2 #define ONE '1'
3 #define TWO "11"
4 #define THREE "111"
5 #define FOUR "1V"
6 #define NONE '-'
7
8 int main(){
9
10     /*Write your code*/
11     int x,y;
12     scanf("%d %d",&x,&y);
13     if(x > 0 && y > 0){
14         printf("%c",ONE);
15     }
16     else if(x > 0 && y < 0){
17         printf("%s",FOUR);
18     }
19     else if(x < 0 && y > 0){
20         printf("%s",TWO);
21     }
22     else if(x < 0 && y < 0){
23         printf("%s",THREE);
24     }
25     else{
26         printf("%c",NONE);
27     }
28     return 0;
29 }
```

## 실습 2

Q. 연도를 나타내는 양의 정수 N을 입력받아, 윤년인지 판별하는 프로그램을 작성해보자.

윤년에 대한 규칙은 다음과 같다.

- ① 주어진 연도가 400으로 나누어 떨어지면 윤년이다.
- ② 주어진 연도가 100으로 나누어 떨어지면 평년이다.
- ③ ②의 조건을 만족하지 않는 경우, 4로 나누어 떨어지면 윤년이다.
- ④ 이외에는 모두 평년이다.

정수 N ( $1 \leq N \leq 2050$ )을 입력받아 윤년이면

`true` 를, 평년이면 `false` 를 출력하자. 단, 입력값이 범위를 넘어가면 계속해서 입력을 받는다.

```
1  #include<stdio.h>
2
3  int main(){
4      int N;
5      /*Write your code*/
6      scanf("%d", &N);
7      while (N < 1 || N > 2050) {
8          getchar();
9          scanf("%d", &N);
10     }
11
12     if (N % 400 == 0) {
13         printf("true\n");
14     } else if (N % 100 == 0) {
15         printf("false\n");
16     } else if (N % 4 == 0) {
17         printf("true\n");
18     } else {
19         printf("false\n");
20     }
21
22     return 0;
23 }
```

## 실습 3

Q. 알파벳 문자들을 입력받아 첫문자는 대문자로,  
나머지는 소문자로 변환하여 출력해보자.

`getchar()` 와 `putchar()` 를 사용하여 아래 조건에  
따라 동작하도록 한다.

- 입력받은 문자중 알파벳 문자가 아닌경우 무시하고  
나머지를 출력한다.
- `\n` 이 입력될때까지 입력받는다.
- 알파벳 문자 : ['a'-'z'], ['A'-'Z']

```
1  #include<stdio.h>
2
3  int main(){
4      /*write your code*/
5      char a;
6      int first = 1;
7
8      while((a = getchar()) != '\n') {
9          int big = a >= 65 && a <= 90;
10         int small = a >= 97 && a <= 122;
11
12         if (!big && !small)
13             continue;
14
15         if (first) {
16             if (big)
17                 putchar(a);
18             else
19                 putchar(a - 32);
20             first = 0;
21         } else {
22             if (big) {
23                 putchar(a + 32);
24             } else {
25                 putchar(a);
26             }
27         }
28     }
29
30     return 0;
31 }
32
```

## 실습 4

Q. 대문자 알파벳을 입력받아, ASCII표에 대응하는 십진수값에 따라 grade를 매겨보자.

점수표 및 수행 조건은 아래와 같다.

점수표

Grade	Score
A	90점 이상
B	80점 이상
C	70점 이상
D	60점 이상
F	60점 미만

조건

- 반드시 `switch()` 문을 사용할 것.
- 대문자 알파벳을 입력할 때까지 입력받을 것.

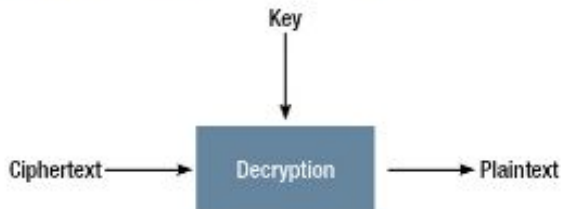
hint : 점수가 10 단위로 커지는 것을 이용할 것

```
#include<stdio.h>

int main(){
    char c;
    int score;
    /*Write your code*/
    do{
        scanf("%c",&c);
    }while(c > 'Z' || c < 'A');
    score = (int)c;
    switch(score/10){
        case 9:
            printf("A"); break;
        case 8:
            printf("B"); break;
        case 7:
            printf("C"); break;
        case 6:
            printf("D"); break;
        default:
            printf("F");
            break;
    }
    return 0;
}
```

# 실습 5

Q. 간단한 암호 해독 프로그램을 만들어보자.



자연수  $N$ 과  $K(1 \leq N, K \leq 26)$ 에 대해서  $N$ 을 *Ciphertext*,  $K$ 를 *Key*라고 하자.

$N$ 과  $K$ 를 입력받아 아래 조건에 따라 *plaintext*를 출력해보자.

조건

- *Plaintext*는 알파벳 'A'를 1번째라고 했을 때,  $N$ 번째 알파벳으로부터  $K$ 값만큼 뒤에 있는 알파벳이다.
- $K$ 값을 더한 순서가 'Z'를 벗어나는 경우는 고려하지 않는다.
- 출력은 대문자 알파벳으로 한다.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

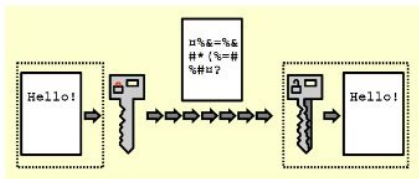
```
1 #include<stdio.h>
2
3 int main() {
4     /*Write your code*/
5     char ciph = 'A';
6     int a, b;
7     scanf("%d %d", &a, &b);
8
9     ciph = ciph + (a - 1) + b;
10    if (ciph <= 'Z')
11        printf("%c\n", ciph); return 0;
12 }
13
```

# Lab. 4 과제 풀이



# 과제 1

Q. 암호화(Encryption) 프로그램을 만들어보자.



`getchar()` 를 이용해 `\n` 전까지 문자를 입력받고, 입력된 문자들을 아래 조건에 따라 암호화 해보자.

조건

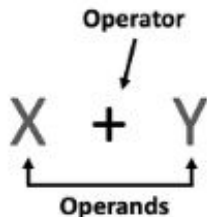
- 첫번째 입력받은 문자를 key로 사용한다. 즉, "ABC" 를 입력했을 경우 'A' 가 key이며, "ABC" 가 plaintext가 된다.
- key값을 2진수로 표현했을 때 1인 자릿수 갯수를 N이라고 할 때, plaintext에 해당하는 모든 문자들을 N만큼 left shift한다.  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- 예를 들어, 'A' 가 3만큼 left shift되면 'X' 가 되며, 'X' 가 2만큼 left shift되면 'V' 가 된다.
- 최소 하나 이상의 알파벳 대문자만 입력된다고 가정한다.

hint : ASCII 코드표를 참고

```
1 #include<stdio.h>
2
3 int main(){
4     /*Writer your code*/
5     int c,cnt=0,n,k;
6     while((c = getchar()) != '\n'){
7         if(cnt == 0){
8             k = (int)c;
9             for(int i=k;i >= 1; i/=2){
10                if(i % 2 == 1){
11                    cnt++;
12                }
13            }
14            n = cnt;
15        }
16        c -= n;
17        if(c < 'A'){
18            c += ('Z' - 'A' + 1);
19        }
20        putchar(c);
21    }
22    return 0;
23 }
```

## 과제 2

Q. 두개의 피연산자(operands)와 하나의 연산자(operator)를 입력받아 계산 값을 출력하는 계산기를 만들어 보자.



조건

- `x + y` 와 같은 형식으로 입력받으며, 각각은 띄어쓰기로 구분한다.  
( $-2,147,483,648 \leq x, y \leq 2,147,483,647$ )
- operator는 `+`, `-`, `*`, `/`, `%`, `<`, `>`만 지원하며, 지원하지 않는 연산자가 입력될 경우 "Invalid operator" 출력.
- operator가 `<` 또는 `>` 일 때는 결과에 따라 `true`, `false` 를 출력.
- Switch문을 사용 할 것.

```
1 #include<stdio.h>
2 #define ERR "Invalid operator"
3
4 int main(){
5     /*Writer your code*/
6     int x,y;
7     char c;
8     scanf("%d %c %d",&x,&c,&y);
9     switch(c){
10        case '+':
11            printf("%d",x+y);
12            break;
13        case '-':
14            printf("%d",x-y);
15            break;
16        case '*':
17            printf("%d",x*y);
18            break;
19        case '/':
20            printf("%d",x/y);
21            break;
22        case '%':
23            printf("%d",x%y);
24            break;
25        case '<':
26            printf("%s",x < y ? "true" : "false");
27            break;
28        case '>':
29            printf("%s",x > y ? "true" : "false");
30            break;
31        default :
32            printf("%s",ERR);
33            break;
34    }
35    return 0;
36 }
```

배열

# 1차원 배열

- 같은 자료형 여러 개를 모아 놓은 구조
  - char 1차원 배열의 경우 문자열을 나타내기 위해 사용될 수 있음.
  - `char arr[5] = "abcW0"` 는 char형 배열 5칸 중 4칸을 사용하고 있는 길이 3짜리 문자열을 나타냄.
  - 문자열은 `printf("%s", arr)` 처럼 %s를 사용하여 출력 가능
  - int, float 등 C언어에서 제공하는 다양한 자료형에 대해 배열을 선언할 수 있음.
    - 나중에 배열 사용자가 직접 정의하는 자료형에 대해서도 배열을 선언할 수 있음.
  - 여러 개의 변수를 일일이 선언하지 않고 묶어서 나타내고 싶을 때 씬.
  - 이번 실습 시간에는 1차원 배열을 사용하도록 하지만 다차원 배열도 존재.

배열에 문자열 입력

# 배열에 문자열 입력 시 주의 사항

- 숫자 등 다른 자료형과 달리, 문자열의 경우 문자열의 길이 + 1 만큼의 크기의 배열이 필요.
  - `char char_array[5]` : 5개의 `char`형 변수 저장 가능
    - `scanf("%s", char_array)`를 통해 길이 4의 문자열 입력 가능
      - 5번째 위치에는 문자열의 끝을 나타내는 'W0' (NULL) 문자 존재 (자동 입력)
      - 따라서 문자열의 길이를 세기 위해서는 'W0' 문자가 나타는 곳을 찾으면 됨!
      - `char_array = "abcW0defW0"`
        - `printf("%s", char_array)` 시 `abc`만 출력 됨.
      - `scanf`에는 항상 주소가 들어가기 때문에 `char_array`의 원소가 아닌, 주소를 보내주어야 함.
        - `scanf("%s", char_array[0])` : X
        - `scanf("%s", &char_array[0])`, `scanf("%s", char_array)` : O

# 과제 / 실습

- 과제 내용
  - 당일 진행한 실습과 관련된 문제
- 제출 기한
  - 실습 : 당일 자정까지, 다음날 자정까지 지각제출 허용 (단, 점수 -30%)
  - 과제 : 다음 실습 수업 전날 23:59 까지
- 제출 방법
  - Elice의 Submit 기능 활용