



# Lab 03

2023학년도 2학기 프로그래밍개론

01, 02, 03분반 TA 유사라  
4ra@sookmyung.ac.kr

---

- 실습: 연습 문제 1-15 & 숫자 처리 문제 1 (최댓값)
- 과제: 숫자 처리 문제 2 (소수 목록)

- 배열 선언

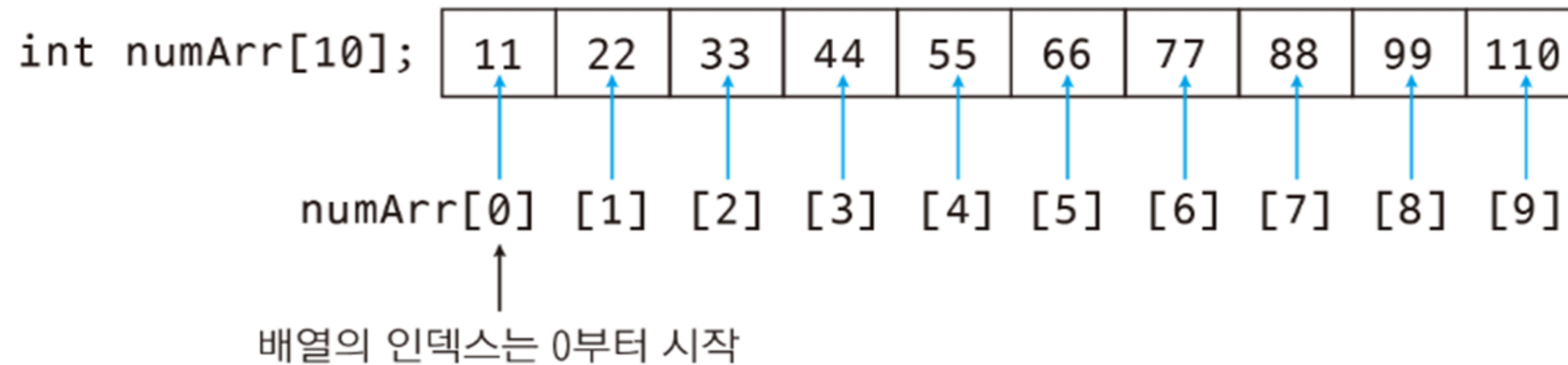
- int numArr[10] : 크기가 10인 정수형 배열 numArr[10] 선언

- 선언 방법

자료형 배열이름[크기];

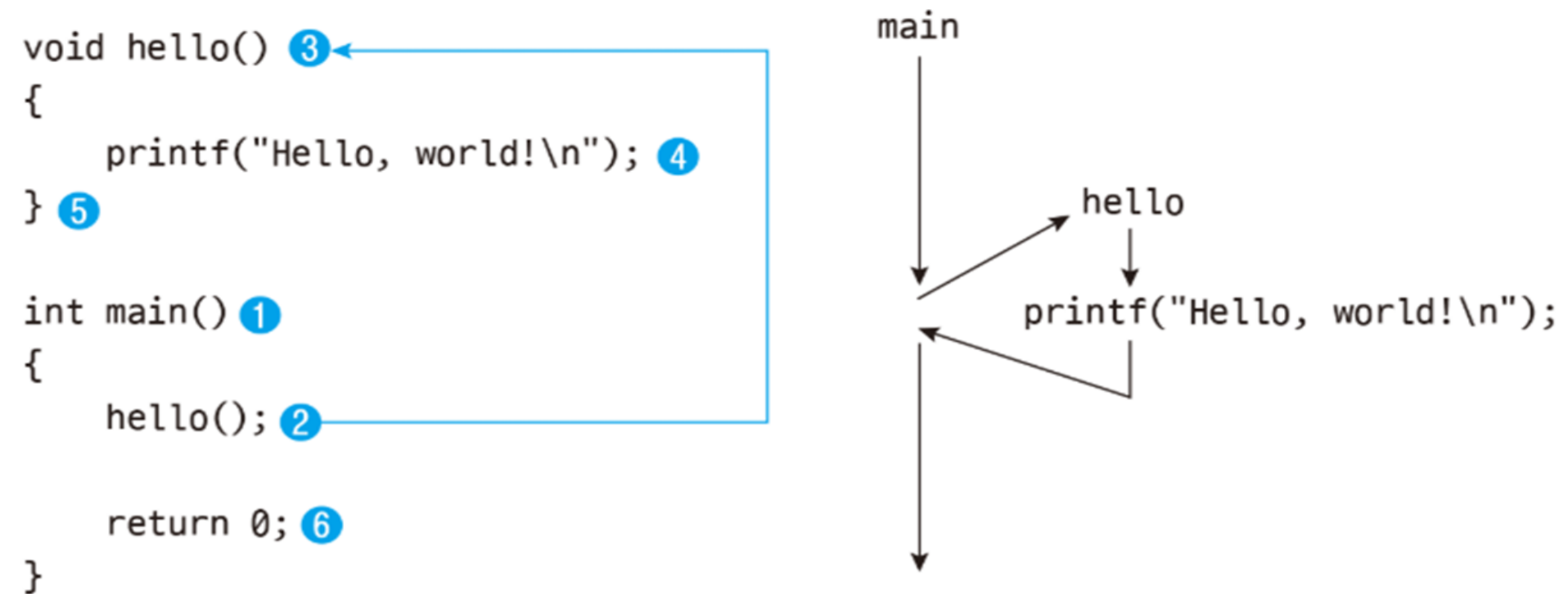
자료형 배열이름[크기] = {값, 값, 값};

- 배열[인덱스] 형태로 요소 접근 가능



- 함수 정의

```
return-type function-name (parameter declarations)
{
    declarations for local variables
    C statements
}
```



- 실습: 연습 문제 1-15

## • 연습 문제 1-15

오른쪽 코드는 1.2절 온도 변환 코드이다.

화씨온도를 섭씨온도로 변환하는 함수 `f2c()`를 작성하여 이용하도록 고친다.

즉, 함수 `f2c()`는 매개변수로 화씨 온도(`fahr`)를 받고, 그에 대응되는 섭씨 온도(`celsius`)를 계산하여 리턴하는 함수이며, `main()`에서는 함수 `f2c()`를 호출하게 될 것이다.

```
#include <stdio.h>
/* print Fahrenheit-Celsius table for fahr = 0, 20, ..., 300,
   floating-point version */
main() {
    float fahr, celsius;
    int    lower, upper, step;
    lower = 0;
    upper = 300;
    step = 20;
    fahr = lower;
    while (fahr <= upper) {
        celsius = (5.0 / 9.0) * (fahr - 32);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

## • 연습 문제 1-15

오른쪽 코드는 1.2절 온도 변환 코드이다.

① 화씨온도를 섭씨온도로 변환하는 함수 f2c()를 작성하여 이용하도록 고친다.

즉, 함수 f2c()는 매개변수로 화씨 온도(fahr)를 받고, 그에 대응되는 섭씨 온도(celsius)를 계산하여 리턴하는 함수이며, main()에서는 함수 f2c()를 호출하게 될 것이다.

① : 함수 기능 → 함수 안에서 수행되어야 할 것

② : 함수 명 → f2c

③ : 파라미터 → 화씨온도(fahr)

④ : 반환값 → 화씨온도(fahr)에 대응되는 섭씨온도 값

```
float f2c (float fahr)
{
    return (5.0 / 9.0) * (fahr - 32);
}
```

## • 연습 문제 1-15

오른쪽 코드는 1.2절 온도 변환 코드이다.

화씨온도를 섭씨온도로 변환하는 함수 **f2c()**를 작성하여 이용하도록 고친다.

즉, 함수 **f2c()**는 매개변수로 화씨 온도(fahr)를 받고, 그에 대응되는 섭씨 온도(celsius)를 계산하여 리턴하는 함수이며, **main()**에서는 함수 **f2c()**를 호출하게 될 것이다.

```
#include <stdio.h>
float f2c(float fahr) {
    return (5.0 / 9.0) * (fahr - 32);
}

main() {
    floatfahr, celsius;
    intlower, upper, step;
    lower = 0;
    upper = 300;
    step = 20;
    fahr = lower;
    while (fahr <= upper) {
        celsius = f2c(fahr);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```



- 숫자 처리 문제 1 (최댓값)

- 숫자 처리 문제 1 (최댓값)

[이 문제에서 배열의 사용은 금지됨]

개수 미정의 정수들을 키보드에서 입력 받아 그 중 가장 큰 수를 출력하는 프로그램을 작성하라. (데이터의 끝은 Ctrl+Z로 표시된다.)

## <의사 코드>

maxval = 첫 번째로 입력을 받은 정수

while (정수 x를 읽을 수 있음)

x가 maxval보다 클 경우(if  $x > \text{maxval}$ ), x가 새로운 maxval 값이 된다 ( $\text{maxval} = x$ )

- 숫자 처리 문제 1 (최댓값)
- [이 문제에서 배열의 사용은 금지됨] → 입력 받은 수를 그때그때 바로 처리해야 함.
- 데이터의 끝은 Ctrl+Z로 표시 → while문의 조건 : Ctrl+Z인 EOF이 들어올 때까지 while문 반복
- 개수 미정의 정수들을 키보드에서 입력받아 그중 가장 큰 수를 출력하는 프로그램을 작성하시오.
  - case1) 새로 입력 받은 수가 기존 maxval 보다 크면 갱신 :  $\text{maxval} = \text{새로 입력받은 수}$
  - case2) 새로 입력 받은 수가 기존 maxval 보다 작다면 그대로 진행

## <의사 코드>

maxval = 첫 번째로 입력을 받은 정수

while (정수 x를 읽을 수 있음)

x가 maxval보다 클 경우(if  $x > \text{maxval}$ ), x가 새로운 maxval 값이 된다 ( $\text{maxval} = x$ )

```
#include <stdio.h>

/* 개수 미정의 정수들을 키보드에서 입력 받아 그 중 가장 큰 수를 출력하는 프로그램 */

main()
{
    int x, maxval;
    scanf("%d", &maxval);           // 변수 maxval 입력받은 수로 초기화
    while (scanf("%d", &x) != EOF) { // EOF(Ctrl+z) 나올 때까지 입력받음
        if (x > maxval) {           // x(새로 입력받은 수) > maxval(현재까지 중 가장 큰 수) 이면
            maxval = x;              // maxval을 x(새로 입력 받은 수)로 업데이트
        }
    }
    printf("maximum value = %d\n", maxval); // 최종 maxval 출력
}
```

- 과제: 숫자 처리 문제 2 (소수 목록)

## 과제: 숫자 처리 문제 2 (소수 목록)

- 1000 미만의 모든 소수를 출력하는 프로그램을 다음과 같은 방법으로 작성하라.

(다른 소수 판별 알고리즘을 이용할 경우 0점 처리)

- 크기가 1000인 배열 `prime[1000]`을 정의한다.
- 배열 `prime[1000]`의 값들을 모두 1로 초기화하며, `prime[p] = 1`은 정수 `p`가 소수임을 의미한다.  
(단, `prime[0] = prime[1] = 0`)
- 가장 작은 소수 2에서 시작하여 (`p`라고 부르기로 함) 각 소수 `p`의 배수들에 대해 소수가 아니라면 해당 `prime[]` 원소 값을 0으로 바꾸어 소수가 아님을 표시한다.
- `p`의 값이 1000의 제곱근인 31.62를 넘을 필요가 없음을 유의한다.
- `prime[]`에서 값이 1로 남아 있는 원소들의 인덱스만 출력하되, 한 줄에 15개씩 아래와 같이 출력한다.

```
2    3    5    7   11   13   17   19   23   29   31   37   41   43   47
53   59   61   67   71   73   79   83   89   97  101  103  107  109  113
...
877  881  883  887  907  911  919  929  937  941  947  953  967  971  977
983  991  997
```

```
prime[0] = prime[1] = 0
for p = 2, ..., 999
    prime[p] = 1
for p = 2, ..., 31
    if prime[p] is 1
        set prime[2*p], prime[3*p], prime[4*p], ... to 0
print p's such that prime[p] is 1
```

<의사 코드>

# 과제: 숫자 처리 문제 2 (소수 목록) - hint

- 1000 미만의 모든 소수를 출력하는 프로그램을 다음과 같은 방법으로 작성하라.  
(다른 소수 판별 알고리즘을 이용할 경우 0점 처리)
- 크기가 1000인 배열 prime[1000]을 정의한다. → 변수 선언 및 초기화
- 배열 prime[1000]의 값들을 모두 1로 초기화하며, prime[p] = 1은 정수 p가 소수임을 의미한다. → 첫 번째 for문  
(단, prime[0] = prime[1] = 0)
- 가장 작은 소수 2에서 시작하여 (p라고 부르기로 함) 각 소수 p의 배수들에 대해 소수가 아니라면 해당 prime[] 원소 값을 0으로 바꾸어 소수가 아님을 표시한다. → 두 번째 for문
- p의 값이 1000의 제곱근인 31.62를 넘을 필요가 없음을 유의한다.
- prime[]에서 값이 1로 남아 있는 원소들의 인덱스만 출력하되, 한 줄에 15개씩 아래와 같이 출력한다.

```
2    3    5    7   11   13   17   19   23   29   31   37   41   43   47
53   59   61   67   71   73   79   83   89   97  101  103  107  109  113
...
877  881  883  887  907  911  919  929  937  941  947  953  967  971  977
983  991  997
```

→ 세 번째 for문



## 과제: 숫자 처리 문제 2 (소수 목록) - hint

- 가장 작은 소수 2에서 시작하여 (p라고 부르기로 함) 각 소수 p의 배수들에 대해 해당 prime[] 원소 값을 0으로 바꾸어 소수가 아님을 표시한다.
- p의 값이 1000의 제곱근인 31.62를 넘을 필요가 없음을 유의한다.

```
for (p는 2부터 p*p가 1000보다 작을 때까지 반복) {  
    // p가 소수가 아니면 prime[p]=0으로 바꿈  
    // 진짜 prime[p]들이 소수인지 확인해야함  
    if(현재 p가 소수라면prime[p]가 소수라면){  
        // 현재 p가 소수인 것 확인하면 됨  
        for(1000보다 작을 때까지 p를 2부터 곱함) {  
            // p가 소수면 p의 배수들은 모두 소수가 아님  
            p의 모든 배수들은 소수가 아니므로 prime[]을 0으로 바꿈  
        }  
    }  
}
```

## 과제: 숫자 처리 문제 2 (소수 목록) - hint

- prime[]에서 값이 1로 남아 있는 원소들의 인덱스만 출력하되, 한 줄에 15개씩 아래와 같이 출력한다.

```
  2    3    5    7   11   13   17   19   23   29   31   37   41   43   47
53   59   61   67   71   73   79   83   89   97  101  103  107  109  113
...
877 881 883 887 907 911 919 929 937 941 947 953 967 971 977
983 991 997
```

→ 세 번째 for문

```
for(p를 0부터 N까지 확인) {
    if(prime[p]가 소수라면) { // 배열의 값이 1일 것임
        %3d 사용하여 print // 숫자 하나 출력 후 띄어쓰기 한번
        변수 i 카운트 증가 // 한 줄에 15개씩 출력하는지 확인하기 위해 사용
        if(i가 15번째면 i는 15의 배수){
            print문으로 줄바꿈 문자(\n) 추가
        }
    }
}
```

- 과제 제출 기한
    - 01, 02분반 : 10월 03일 (화) 23:59까지 스노보드로 제출
    - 03분반 : 10월 04일 (수) 23:59까지 스노보드로 제출
    - 제출 기한 이후 24시간 이내에 지각 제출할 경우 : 2점 감점
    - 그 이후는 0점 처리(추가 제출 받지 않음)
  - 제출 장소
    - 스노우보드 과제 제출 페이지에 업로드
  - 소스파일(.c)과 과제보고서(.docx)가 담긴 압축파일(.zip) 제출
    - 압축 파일 이름: Lab03\_학번\_이름.zip  
'Lab03\_학번\_이름'으로 된 c파일 + 'Lab03\_학번\_이름'으로 된 .docx파일
  - 과제보고서(.docx) 양식
    - 스노우보드에서 다운로드
- 1) 실행 결과 화면 캡처한 이미지 2) 소스코드 3)설명 작성

- 조교 메일로 질문 보내기
  - [4ra@sookmyung.ac.kr](mailto:4ra@sookmyung.ac.kr)

- 질문시 주의사항

- 메일에 반드시 과목, 분반, 전공, 이름, 학번 명시

제목 : [프로그래밍개론 001분반] 2331297 유사라 Lab01 질문 드립니다.
- 몇 번 과제에서 어떤 부분이 막혔는지, 어떤 부분이 문제인지 코드와 함께 설명 첨부  
(그냥 코드만 보내면 어디가 문제인지 알 수 없어요)
- 답장이 늦을수도 있으니 이 점 고려하여 미리 질문 (특히 과제 제출 마지막날 유의!)
- 질문 내용을 구체적으로 명확하게 적어 주시기 바랍니다.
- 오류 메시지를 첨부하고 싶을 경우, 오류 캡처 화면 + 전체 코드 c 파일을 첨부하여 보내주세요. (코드 캡처 사진 X)
- 그 외 출석 등 다른 질문들도 메일로