
프로그래밍을통한
논리적사유연습

제 1 장

소프트웨어 교육과

C언어의 시작

문제 해결을 위한 '컴퓨터 과학적 사고'

- **컴퓨터 과학적 사고(Computational Thinking)**
  - 컴퓨터가 문제를 해결하는 방식처럼 복잡한 문제를 단순화하고 이를 논리적, 효율적으로 해결하는 사고능력
  - 우리가 실생활에서 겪는 여러 문제를 컴퓨터가 일을 처리하는 것처럼 논리적으로 해결

- 일상 생활속의 Computational Thinking
  - 우선순위를 고려한 서비스
  - 은행, 슈퍼마켓, 입국심사에 줄서기
  - 최단경로 찾기나 최단거리 일주
  - 특정 물건 또는 정보 찾기
  - 트럭 짐 싣기
  - 등…

키워드 출처: http://terms.naver.com/entry.nhn?docId=3607507&cid=58598&categoryId=59316/

알고리즘이란?

- 알고리즘(algorithm)
  - 어떤 주어진 문제를 논리적으로 해결하기 위해 구성된 일련의 순서화된 절차, 방법, 명령어들을 모아놓은 것

- 알고리즘의 조건
  - 0개 이상의 입력과 1개 이상의 출력
  - 종료되어야 함
  - 모든 명령이 실행 가능해야 함

키워드 출처: http://terms.naver.com/entry.nhn?docId=814914&cid=42344&categoryId=42344

알고리즘이란?

- 정렬 알고리즘(sort algorithm)
  - 데이터를 일정한 규칙에 따라 재배열하는 것을 의미

15	11	1	3	8

- 오름차순 정렬(작은 수부터 큰 수의 순으로 나열)

1	3	8	11	15

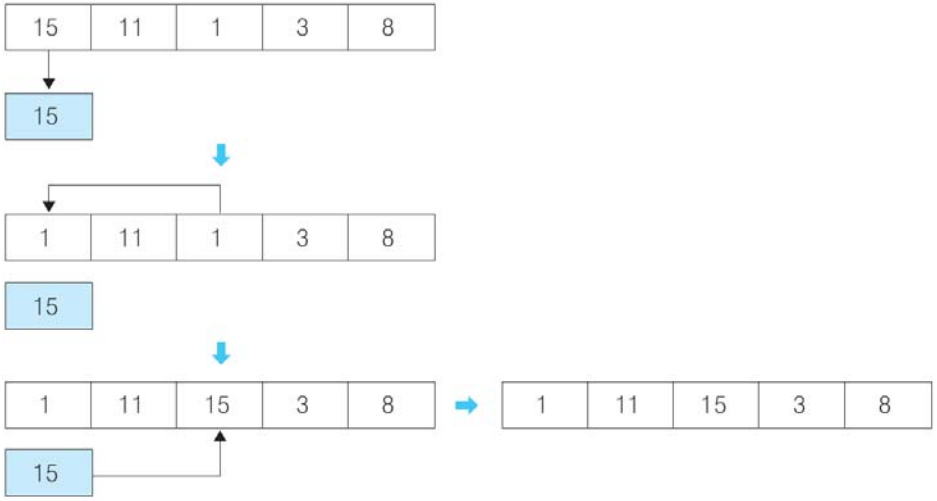
- 내림차순 정렬(큰 수부터 작은 수의 순으로 나열)
- 정렬 알고리즘 선택 정렬, 삽입 정렬, 버블 정렬, 퀵 정렬, 병합 정렬, 힙 정렬 등

알고리즘이란?

- 선택 정렬
  - 정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 가장 앞의 데이터와 교환해나는 방식

15	11	1	3	8

① 가장 작은 데이터인 1을 가장 앞에 위치한 15와 교환. 가장 작은 데이터가 가장 앞에 위치.

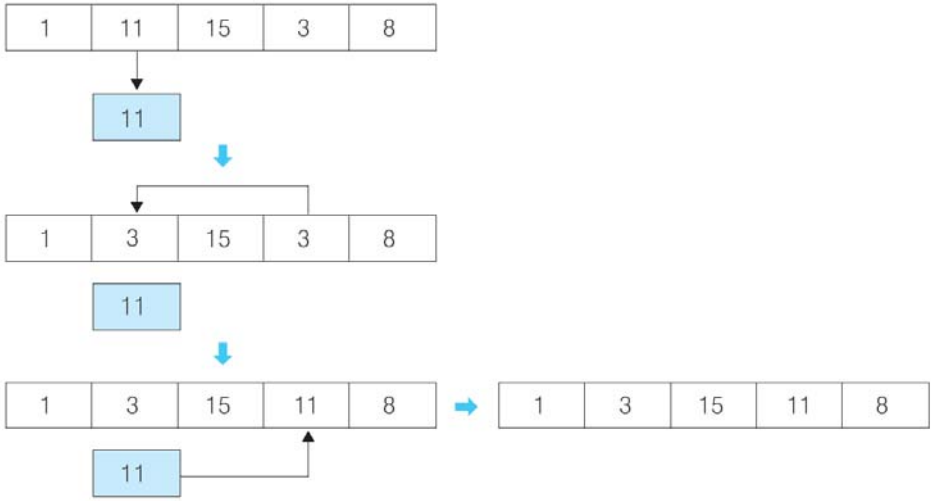


알고리즘이란?

- 선택 정렬
  - 정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 가장 앞의 데이터와 교환해나는 방식

15	11	1	3	8

② 첫 번째 데이터를 제외한 나머지 데이터에서 가장 작은 데이터인 3을 두 번째 데이터인 11과 교환

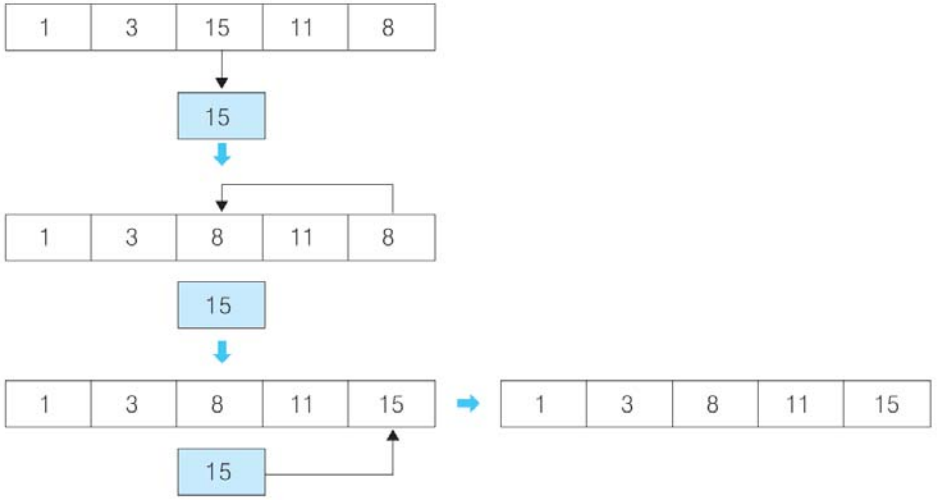


알고리즘이란?

- 선택 정렬
  - 정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 가장 앞의 데이터와 교환해나는 방식

15	11	1	3	8

③ 첫 번째, 두 번째 데이터를 제외한 나머지 데이터에서 가장 작은 데이터인 8을 세 번째 데이터인 15와 교환

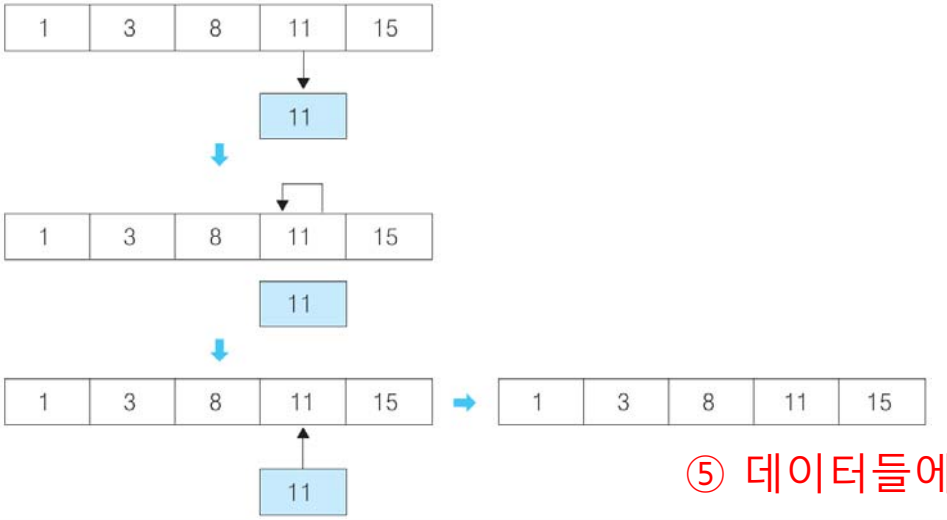


알고리즘이란?

- 선택 정렬
  - 정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 가장 앞의 데이터와 교환해나는 방식

15	11	1	3	8

④ 첫 번째, 두 번째, 세 번째 데이터를 제외한 나머지 데이터에서 가장 작은 데이터인 11을 네 번째 데이터인 11과 교환. 같은 데이터이므로 위치의 변화는 없음.



⑤ 데이터들에 대한 정렬이 완료

알고리즘이란?

```
#include <stdio.h>
#define MAX 20  /* 데이터의 최대 수를 의미하는 매크로 */

void select_sort (int data[], int n);

main()
{
   int i, n;
   int data[MAX];
   printf("number of data => ");
   scanf("%d", &n);
   for (i=0; i<n; i++) {
      printf("%i\'s number => ", i+1);
      scanf("%d", &data[i]);
   }
   select_sort (data, n);  /* select_sort 함수를 호출 */
   printf("\nsorted data : ");
   for (i=0; i<n; i++) {
      printf("%d ", data[i]);
   }
   printf("\n");
}

void select_sort (int data[], int n)
{
   int i, j, k, temp;
   for (i=0; i<n-1; i++) {
      k = i;
      /* 가장 작은 값을 지닌 데이터(data[k])를 찾아냄 */
      for (j=i+1; j<n; j++) {
         if (data[k] > data[j])
            k = j;
      }
      temp = data[i];
      data[i] = data[k];
      data[k] = temp;
   }
}
```

프로그래밍이란? 코딩이란?

- 프로그램이란?
  - 컴퓨터를 실행시키기 위해 차례대로 작성된 명령어 모음.

- 프로그래밍이란?
  - 작업을 컴퓨터에 알맞도록 정리해서 순서를 정하고 컴퓨터 특유의 명령코드로 고쳐 쓰는 작업을 총칭해서 프로그래밍
  - 컴퓨터의 명령 코드를 쓰는 작업을 특히 코딩(coding)이라고도 함

- 프로그램 작성 과정

키워드 출처: http://terms.naver.com/entry.nhn?docId=1159441&cid=40942&categoryId=32837

프로그래밍이란? 코딩이란?

1단계 문제의 정의 및 분석	프로그램의 목적, 사용자, 입출력, 데이터 처리에 관한 요구사항, 구현상의 문제 분석, 타당성 조사 등 전과정에 대한 문서화
2단계 알고리즘 설계	계층 차트, 순서도 또는 유사 코드(pseudocode)를 사용해서 프로그램의 논리적인 실행 순서와 내용을 구체화
3단계 프로그램 코딩	적절한 고급 언어를 선택해서 구문에 맞는 프로그램 작성
4단계 프로그램 테스팅	프로그램 디버깅(알파테스팅)과 실질적인 데이터와 실질적인 사용자에 의한 시험 실행(베타테스팅)
5단계 프로그램 문서화/유지 보수	하드웨어와 소프트웨어 요구사항, 입출력과 프로그램 파일 관리 등 전체적인 사용 설명서 작성, 프로그램의 효과적인 사용을 위한 유지 및 보수

C 언어란?

- C 언어의 기원과 역사
  - 1972년 AT&T사의 Bell 연구소에서 UNIX를 개발할 수 있는 고급 언어로 개발됨
  - BCPL 언어의 부족한 점을 보강하기 위해 켄 톰슨(Ken Thompson)이 B 언어 개발
  - 데니스 리치(Dennis Ritch)가 B언어를 C 언어로 발전시킴
  - UNIX의 70% 이상을 담당하는 고급 언어
  - 지금까지 많은 개발 환경에서 사용됨

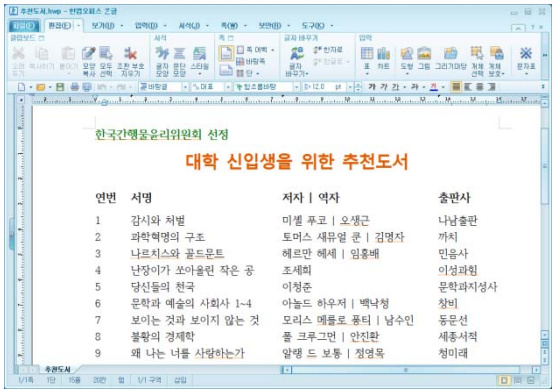
C 언어란?

- C 언어의 특징
  - 시스템 프로그래밍 언어
  - 함수 언어
  - 강한 이식성
  - 풍부한 자료형 지원
  - 다양한 제어문 지원
  - 표준 라이브러리 함수 지원

프로그램 실행 과정

- 컴퓨터의 구성

(a) 하드웨어



한국간행물윤리위원회 선정

대학 신입생을 위한 추천도서

연번	서명	저자 | 역자	출판사
1	감시와 처벌	미셸 푸코 | 오생근	나남출판
2	과학혁명의 구조	토머스 새뮤얼 쿤 | 김명자	까치
3	나르치스와 골드문트	헤르만 헤세 | 임홍배	민음사
4	난장이가 쏘아올린 작은 공	조세희	이성과힘
5	당신들의 천국	이청준	문학과지성사
6	문학과 예술의 사회사 1~4	아놀드 하우저 | 백낙청	창비
7	보이는 것과 보이지 않는 것	모리스 메를로 퐁티 | 남수인	동문선
8	불황의 경제학	폴 크루그먼 | 안진환	세종서적
9	왜 나는 너를 사랑하는가	알랭 드 보통 | 정영목	청미래

(b) 소프트웨어

프로그램 실행 과정

- 하드웨어
  - 전자회로와 물리적인 장치로 이루어진 것

- 하드웨어의 5대 구성 요소
  - 입력장치 : 데이터를 컴퓨터가 이해하도록 전기 신호로 변환함
  - 중앙처리장치 : 컴퓨터의 구성요소를 제어하거나 연산을 수행함
  - 출력장치 : 중앙처리장치가 처리한 결과를 출력함
  - 주기억장치 : RAM. 컴퓨터 내에서 실행중인 명령어와 데이터를 저장함
  - 보조기억장치 : 주기억장치를 보조해주는 장치. CD-ROM, 외장하드, usb 등이 있음

프로그램 실행 과정

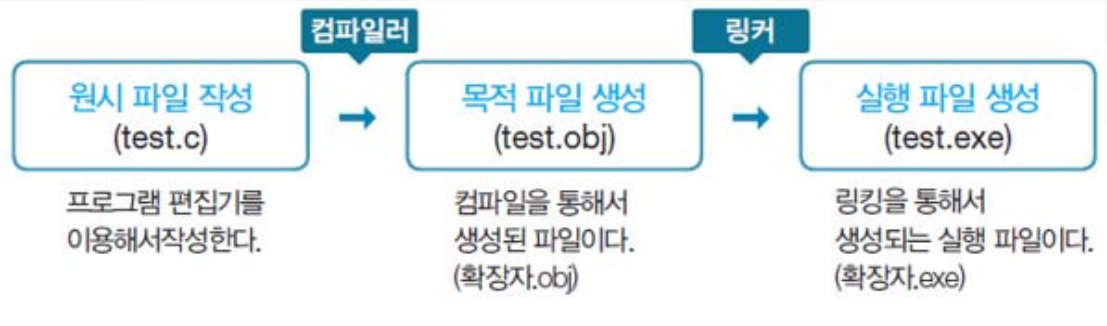
- 소프트웨어
  - 하드웨어에 명령을 내려 구동시킬 수 있도록 만들어진 프로그램
  - 명령어들의 집합
  - 시스템 소프트웨어와 응용 소프트웨어로 나뉨

- 시스템 소프트웨어
  - 컴퓨터를 효율적으로 운영하고 제어하기 위한 프로그램
  - Ex) 도스(DOS)나 윈도우(Windows) 같은 운영체제, 여러 유틸리티

- 응용 소프트웨어
  - 특정 분야의 업무를 처리하기 위한 프로그램
  - Ex) 한글이나 워드, 포토샵, 엑셀 등

프로그램 실행 과정

- 프로그램의 실행 절차



  - 원시 파일
    - 사용자가 편집기(메모장이나 워드패드)를 이용해서 만든 파일
  - 컴파일러
    - 사용자가 만든 원시 파일을 기계어로 바꿔줌
    - 목적 파일의 확장자는 .obj(object)
  - 링커 (Linker)
    - 소스코드와 시스템 라이브러리 파일을 연결(linking)시킴
    - 목적 파일 여러 개를 연결시켜 실행 파일 1개를 생성

C 프로그램의 구조

- C 프로그램의 구조
  - 1개 이상의 함수로 구성됨
  - Main( ) 함수는 반드시 존재해야 함
  - 그 외의 함수는 사용자가 임의로 작성해서 사용할 수 있음

```
// 전처리기
#include <stdio.h>                    ①

int main(void)                        ②
{                                     ③
   // 일반 명령문들                    ④
   .....
   return 0; // main() 함수의 반환    ⑤
}
```

C 프로그램의 구조

① 전처리기 지시자
- 항상 #으로 시작하고, 한 줄에 지시자 1개만 쓸 수 있음
- 헤더파일을 포함시키거나 상수 및 간단한 매크로를 정의할 때 사용함
- 문장의 마지막에 세미콜론(;)을 사용하지 않음

② main(void) 함수
- 프로그램의 시작을 알리는 함수
- C 프로그램은 main( ) 함수 1개와 하위함수로 구성됨

③ 중괄호 { }
- 명령문들의 블록(block)

```
// 전처리기
#include <stdio.h>                    ①

int main(void)                        ②
{                                     ③
   // 일반 명령문들                    ④
   .....
   return 0; // main() 함수의 반환     ⑤
}
```

C 프로그램의 구조

④ 일반 명령문들
- 실제로 프로그램이 실행할 기능들을 실행 순서대로 기술해놓은 부분
- 실행 순서는 위 → 아래, 왼쪽 → 오른쪽 이며, 경우에 따라 바꿀 수 있음

⑤ 함수의 반환
- return문으로 함수의 실행 결과를 반환함

```
// 전처리기
#include <stdio.h>                    ①

int main(void)                        ②
{                                     ③
   // 일반 명령문들                    ④
   .....
   return 0; // main() 함수의 반환     ⑤
}
```