

Study Guide for Algorithm Mid-Term Exam

1. 교과목 목표 :

- 컴퓨터로 문제해결 하는 과정을 설계하여 프로그래밍 기법 향상
- 고급자료구조와 프로그래밍 논리를 익혀 스스로 프로그래밍 하는 능력 향상
- 이미 알려진 유명한 알고리즘의 개발 및 구현과정의 이해를 통하여 프로그래밍의 원리이해

2. 수업 내용 및 평가 : 강의계획서 참조

자료구조를 바탕으로 문제에 대한 해결책을 구상하고 C로 구현한다.

- 컴퓨팅사고와 기본개념(1주)
- 알고리즘의 정의, 표현, 제어(순차, 조건, 반복)
예제를 통한 알고리즘의 이해 및 작성연습(2주)
- 알고리즘을 프로그래밍 언어로 구현
C의 기본 자료처리와 함수 작성에 의한 모듈화(3주)
- Recursive Algorithm(gcd, Hanoi)
반복제어구조와 recursio의 이해(4주)
- 알고리즘 설계와 구현단계 : Top-down 방식의 문제해결절차 연습
Selection Sort(선택정렬)(5주)
Binary Search(이진검색)(6주)
- 규칙을 코드로 구현(Magic Square)(6주)
- 기타 : 배열, 파일처리, 구조체의 자료구조활용/컴퓨팅사고-분해/추상화/규칙찾기

평가 : 중간(30), 기말(30), 과제물및기타(20), 출석(20)

개념정리

1. 다음 개념의 핵심을 정리하시오

- (1) 알고리즘의 5가지 조건을 쓰시오.
- (2) 보편적인 프로그래밍 문장의 종류를 쓰시오.
- (3) 알고리즘 설계와 구현단계를 쓰시오.
- (4) 함수원형에 포함되는 정보를 쓰시오
- (5) 배열을 함수의 파라미터로 줄 때 전달하는 데이터가 무엇인지 쓰시오
- (6) n 개의 디스크를 가진 하노이타워 문제에서 몇 번의 움직임이 있는지 쓰시오
- (7) 오일러의 경로가 존재하는 정점 6개의 그래프의 예를 하나 그리고,
시작 정점에서 마지막 정점까지의 경로를 쓰시오

2. 다음 용어를 영어로 적으시오

들여쓰기/주석/일반화/변수/상수/배열/재귀/알고리즘/자료구조/
유한성/정확성/구문/구현/선언문/함수원형

알고리즘작성과 함수 구현

1. 수업시간에 한 기본제어를 활용한 예제-수업정리과제 참고

2. 열이 2개인 이차원 배열문제

3. 함수작성-C함수 구현

예제) n명 학생id와 그 학생의 점수데이터를 열이 2개인
2차원 배열 scores에 준비하고 있다고 가정할 때,
어떤 학생 sid를 입력으로 주면 그 점수를 리턴해 주는
함수 find_jumsu를 작성하시오.

```
int find_jumsu(int scores[][2], int n, sid {  
    int k;  
    for (k = 0; k < n; k++)  
        if (scores[k][0] == sid)  
            return scores[k][1];  
    return 0;  
}
```

재귀용법 - 유클리드알고리즘

최대공약수를 구하는 다음의 정의에 따라

$\text{gcd}(72, 16)$ 을 구하는 동안 재귀적으로 호출되는 과정과

그 결과인 72와 16의 최대공약수를 쓰시오

$$\text{gcd}(x, y) = x, \text{if } y=0$$

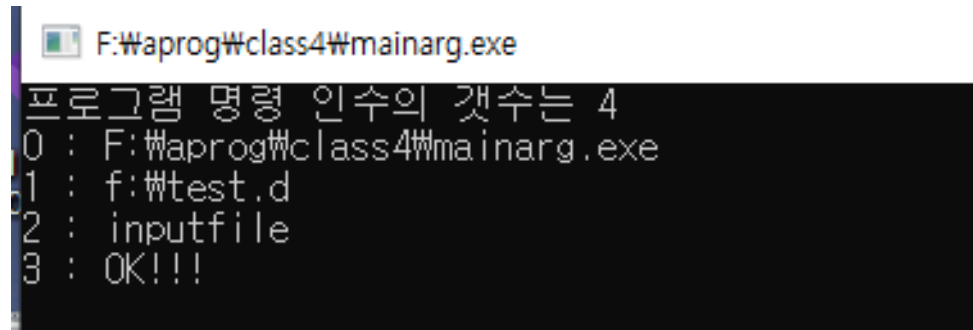
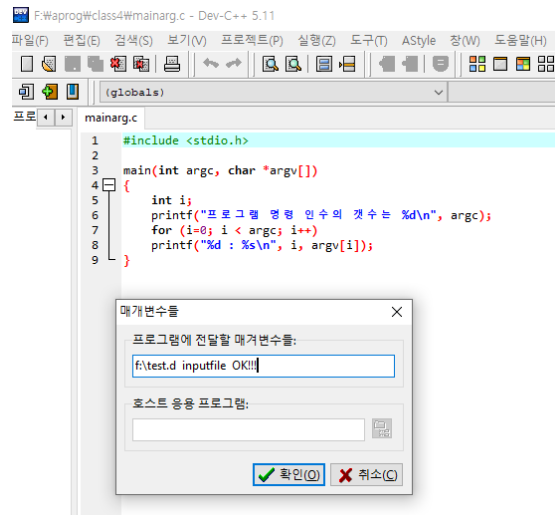
$$= \text{gcd}(y, x \% y), \text{otherwise}$$

외부 파일 처리

다음 프로그램의 명령인수(실행매개변수값)로

f:\test.d inputfile OK!!! 이 주어진 경우 다음 프로그램에서 출력하는 것을 쓰시오.

```
main(int argc, char *argv[])
{
    int i;
    printf("프로그램 명령 인수의 갯수는 %d\n", argc);
    for (i=0; i < argc; i++)
        printf("%d : %s\n", i, argv[i]);
}
```



선택정렬

다음은 배열 `a`에 데이터를 오름차순으로 정렬하기 위한 선택정렬(selection sort) 프로그램의 핵심적인 부분을 C 코드로 나타낸 것이다.

밑줄 친 곳에 알맞은 조건문을 쓰시오.

```
for (s=0; s < n-1; s++) {
```

```
    _____  
    for (j=s+1; j < n; j++)
```

```
        _____  
        temp = a[s]; a[s] = a[m]; a[m] = temp;
```

파일과 구조체활용

어떤 학생의 아이디(id), 이름(name), 성적(gpa)을 다음과 같이 선언된 구조체 타입 example의 배열에 저장하고 있다고 하자. 이때 저장된 n개의 데이터 중에 최대 score를 가진 데이터example을 리턴하는 함수 find_max()를 작성하시오

```
typedef struct member {  
    int id;  
    char name[20];  
    float score;  
} example;  
example find_max(example *a, int n);
```

```
example find_max(example a[], int n)  
{  
    int k, max_index;  
    max_index = 0;  
  
    for (k=1; k < n; k++)  
        if (a[k].score > a[max_index].score)  
            max_index = k;  
    return a[max_index];  
}
```

이진검색

다음은 데이터가 내림차순으로 저장되어 있는 경우 이진검색(binary search) 프로그램을 작성하기 위한 과정이다. 다음 물음에 답하시오.
구간을 표시하기 위한 변수로서 left와 right를, 중앙값의 배열 index로 mid를 사용한다고 하자.

(1) 예제로서 다음의 데이터를 가지고 시뮬레이션하는 과정의 표를 채우시오.
list[]={150, 119, 105, 90, 75, 65, 50, 45, 35, 25, 10}, key = 100

left	right	mid

(2) (1)의 시뮬레이션 과정에서 중앙값과 비교하며 구간을 줄여가는 과정을 구간 안에 데이터가 존재하는 동안 계속 한다.
그 과정을 코드로 나타내었을 때 다음 빈 곳을 채우시오.

```
while ( _____ ) {  
    mid = (left + right) / 2;  
    if (key < list[mid]) )  
    else if (key > list[mid])  
    else return mid;  
}
```


규칙을 코드로 - 마방진

다음은 magic square를 완성하기 위한 Coxeter의 규칙을 코딩한 내용이다. 이를 이해하고 물음에 답하시오.

```
i = 0; j = (n-1)/2;
table[i][j] = 1;
for (count = 2; count <= n*n; count++) {
    row = (i-1 < 0) ? (n-1) : (i-1); /* up */
    col = (j-1 < 0) ? (n-1) : (j-1); /* left */
    if (table[row][col]) i++;
    else {
        i = row; j = col;
    }
    table[i][j] = count;
}
```

- 1) n 이 3인 경우 count가 6일 때 $i=0$, $j=0$ 이다. 그 다음 스텝으로 count=7인 경우 `table[i][j] = count;`을 수행하는 시점에서 row, col, i, j 의 값은?
- 2) 이 프로그램은 table을 모두 0으로 초기화해야 제대로 동작한다. 그 이유와 초기화한 것을 사용하는 문장은 무엇인가?

중간고사 준비는 이렇게

1. 1~6주차 수업내용 검토 수업자료+수업정리과제
2. 배운 알고리즘 모방하며 익히기
 - 기본 제어 알고리즘 예제
gcd/hanoi/
selection_sort/binary_search/magic square
3. 고급자료구조 활용
 - 배열.구조체,파일 등