

과제1 소개

- 보드 게임 만들기 -

2019.04.01

전자공학 프로그래밍

과제 추진 계획

- 2019-1학기 전자공학프로그래밍 실습 진행 계획

순서	내용	비고
실습1	Visual Studio 설치 및 사용법	2주차
실습2	함수와 배열을 이용한 실습	5주차 (과제1)
실습3	포인터와 함수를 활용한 실습	7주차 (과제2)
실습4	C 프로젝트	9주차 (프로젝트1)
실습5	JAVA 개발 환경 설명 및 클래스를 이용한 프로그래밍	10주차
실습6	상속, 인터페이스, 패키지를 활용한 프로그래밍	12주차 (과제3)
실습7	JAVA 프로젝트	14주차 (프로젝트2)

과제 및 프로젝트 출제 계획

- 2019년도 1학기 과제 및 프로젝트 출제 계획
 - 과제 비율** : 전체 평가의 **30%** (과제 3 + 프로젝트 2)
 - 과제 별 평가 비율

	과제1	과제2	과제3
Language	C	C	JAVA
Scoring	5점	5점	5점

- 프로젝트 별 평가 비율

	프로젝트1	프로젝트2
Language	C	JAVA
Scoring	7점	8점

과제 1. 보드 게임 프로그램

- **보드 게임 프로그램**
보드 게임 규칙
- **보드 게임 프로그램 구현**
프로그램 구조
구현해야 하는 내용
실행 화면(목표 및 구현해야 되는 내용)
- **과제 채점 기준** 소개

보드 게임 프로그램

- 보드 게임 규칙

- 0부터 7까지 숫자가 쓰여있는 5x5 정사각형 보드에서 게임을 한다
 - ▶ 보드에 쓰여있는 숫자는 0부터 7까지의 숫자 중 랜덤
- 보드의 가장 왼쪽 위에 동전을 하나 올려놓고, 다음과 같이 동전을 움직인다
 - ▶ 동전이 있는 곳에 쓰여있는 숫자 α 를 본다
 - ▶ 위, 아래, 왼쪽, 오른쪽 방향 중 한 가지를 고른다
 - ▶ 동전을 위에서 고른 방향으로 α 만큼 움직인다
 - ▶ 위의 동작을 반복한다
- 동전이 보드의 바깥으로 나간다면 게임은 종료된다
- 동전이 있는 곳에 쓰여있는 숫자가 0이어도 게임은 종료된다

보드 게임 프로그램

- 프로그램 구조

```
#define Width 5
#define Depth 5
#define MAX_MOVEMENT 25

int square_board[Width][Depth];
int max_movement_count = 0;

void setBoard(int max_width, int max_depth);
void printBoard(int max_width, int max_depth);
void playBoard(int width, int depth, int movement_count);
```

- 매크로 변수 및 전역 변수
 - ▶ Width, Depth, MAX_MOVEMENT, square_board, max_movement_count
- 함수로 구성
 - ▶ setBoard, printBoard, playBoard

보드 게임 프로그램

- 구현해야 하는 내용
 - 문제 : 랜덤으로 생성되는 보드에서 최대 이동 가능한 횟수를 출력하라
 - 제약사항
 - ▶ **메인 함수**는 수정하지 않는다
 - ▶ 프로그램에서 나타나는 출력 형식과 동일하게 출력되도록 한다
 - 무한번 움직일 수 있다면 -1을 출력한다
 - ▶ 3가지 함수를 사용한다. (보드 생성, 보드 출력, 게임 실행)
 - void setBoard(int max_width, int max_depth);
 - void printBoard(int max_width, int max_depth);
 - void playBoard(int width, int depth, int movement_count);
 - ▶ 최대 이동 횟수와 보드는 전역 변수로 선언된 변수 및 배열을 활용한다
 - int square_board[Width][Depth];
 - int max_movement_count = 0;

보드 게임 프로그램

- 구현해야 하는 내용
 - 문제 : 랜덤으로 생성되는 보드에서 최대 이동 가능한 횟수를 출력하라
 - 함수 기능 명세
 - ▶ void setBoard(int max_width, int max_depth);
 - 5x5 크기의 보드에 랜덤한 숫자를 채워 넣어서 생성
 - ▶ void printBoard(int max_width, int max_depth);
 - 랜덤으로 생성된 보드의 상태를 출력
 - ▶ void playBoard(int width, int depth, int movement_count);
 - width, depth: 동전이 놓여있는 위치를 입력
 - movement_count: 동전이 현재까지 움직인 횟수를 입력

보드 게임 프로그램

- 실행 화면 (목표 및 구현해야 되는 내용)

```
3 2 4 2 5
3 2 7 6 3
5 0 2 4 5
7 7 3 3 1
1 4 5 5 0
```

최대 이동 횟수: -1

```
0 1 0 5 4
1 7 6 3 2
7 7 5 3 0
7 0 2 0 1
1 1 1 2 2
```

최대 이동 횟수: 0

```
7 2 3 0 7
3 3 2 6 0
1 2 0 4 3
6 2 1 3 3
6 0 1 0 1
```

최대 이동 횟수: 1

```
2 4 0 3 4
3 7 4 0 0
7 3 2 1 1
3 5 4 3 5
6 4 3 0 6
```

최대 이동 횟수: 2

```
3 2 6 5 7
0 4 1 1 5
7 3 1 2 5
2 4 7 7 0
0 6 1 7 6
```

최대 이동 횟수: 3

```
3 2 5 7 1
7 6 6 5 4
6 4 4 0 5
4 2 3 6 3
1 0 2 4 4
```

최대 이동 횟수: 6

과제 채점 기준

- Due Date : 4/14(일) 자정 (기한 엄수)

감점	지각	(4/14에서 4/15로 넘어가는 자정까지 제출)	감점 x	블랙보드에 업로드 해주세요. (블랙보드가 문제되는 경우, 메일로 보내주세요.)
		~ 3일 지각	(총점수) * 0.75	총 점수에서 25% 감점
		~ 7일 지각	(총점수) * 0.50	총 점수에서 50% 감점
		7일 이상 지각	0	과제 마감 시점을 기준으로 7일 이후로는 받지 않습니다.
	코드 미제출	보고서 코드가 있을 때,	-30	
		보고서에 코드가 없을 때,	구현점수 x	
	메인 함수 수정		-5	(-5) 감점 후 메인 함수 수정 후 구현 채점기준 적용
	cpp 파일 제출			c 파일로 확장자 변환 후 구현 채점기준 적용

과제 채점 기준

- 평가 기준

- <구현(70%) + 보고서(30%)>

: 구현과 보고서의 평가는 기능별로 평가를 할 예정이며, 구현의 경우는 각 기능을 명시된 설명에 맞게 구현하였는지, 구현하지 못하였다면 자신이 구현한 부분에 대해 어느 정도 설명이 되어 있는지를 보고 평가할 예정입니다

- ✓ **필수** : Visual Studio 2017로 구현 권장
 - ✓ 각 함수에 대한 설명은 주석을 참조하여 구현
 - ✓ 결과 화면에 수행해 본 예시에 대한 분석 필수(보고서)
 - ✓ 프로젝트 제출 x, .cpp 파일 x
 - ✓ 프로젝트 제출 또는 .cpp 파일 제출의 경우 **감점** 예정

과제 채점 기준

- 평가 기준
 - 구현 부분(70%)
 - ▶ 구현 함수 혹은 구현 부분에 대한 **기능 평가** 예정
 - 보고서 부분(30%)
 - ▶ 구현된 함수의 기능 명세
 - ▶ 소스코드 분석
 - ▶ 결과 화면에 대한 분석
 - 모든 보고서 작성은 본인이 직접 작성한 내용이어야 함 (강의노트 paste X)
 - .c 파일과 보고서 파일 제출 (프로젝트 제출 x, .cpp 파일 x)

Q&A

AJOU UNIVERSITY
Embedded & Software Lab.

TA : 임창남, 유동연

E-mail : chn0714@naver.com, dongs0125@ajou.ac.kr

Office : 원천관 **338**호 임베디드소프트웨어 연구실