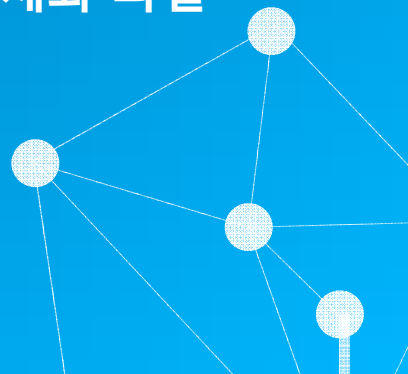




04 CHAPTER 배열, 구조체와 파일



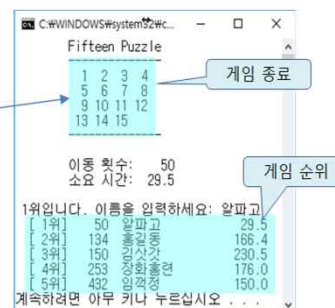
Target: 4x4 퍼즐 게임



- 4x4 퍼즐 게임(fifteen puzzle game)
 - 1~15까지의 숫자 퍼즐을 제자리로 맞추는 게임
 - 퍼즐 섞기, 퍼즐 맞추기
 - 시간 측정
 - 랭킹 처리
 - 파일 입출력

12	15	6	10
4	9	5	8
14	13		2
1	7	11	3

4x4퍼즐게임



4장 학습 목표



- 배열의 개념을 이해하고, 배열의 선언 및 활용 능력을 기른다.
- 함수 호출에서 배열의 전달 방법을 이해한다.
- 문자열이 다른 배열들과 다른 점들을 이해한다.
- 구조체를 이해하고 구조체 객체의 활용 능력을 기른다.
- 구조체와 관련된 함수의 호출과 반환을 이해한다.
- 텍스트 파일과 이진 파일을 이해하고, 관련 함수를 사용하는 능력을 기른다.
- 랭킹 처리를 위한 자료의 정렬 개념을 이해한다.

3

4.1 대용량 자료의 처리



- 대용량 자료의 처리 방법?
- 배열이란?
- 구조체란?
- 배열, 구조체와 파일

4



대용량 자료의 처리 방법?



- 한꺼번에 많은 자료 처리가 필요한 경우
 - 배열
 - 구조체

1번	2번	3번	...	45번
84	72	94	...	87

반 학생들의 성적 처리

홍길동
이름: 홍길동
전화: 010-1234-56XX
주소: 서울특별시 XX구 YY동...
이메일: kdhong@korea...

친구 주소록 만들기

- 데이터를 컴퓨터에 저장하고 싶은 경우
 - 파일(file) 사용
 - 프로그램이 종료되어도 자료가 저장되어 있음

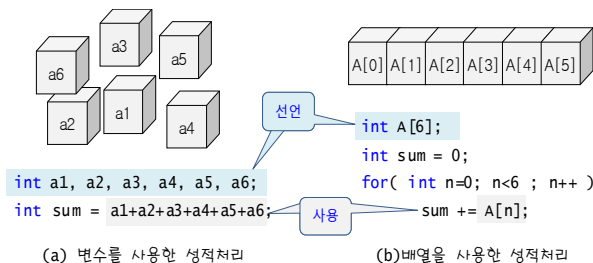
5

배열이란?



- 동일한 자료형의 데이터를 한꺼번에 만들 때 사용
 - 하나의 이름과 인덱스(index) 사용

```
int a1, a2, a3, a4, a5, a6; // 6개의 변수를 각각 선언
int A[6]; // 배열로 선언
```



- 반복문을 사용할 수 있다!

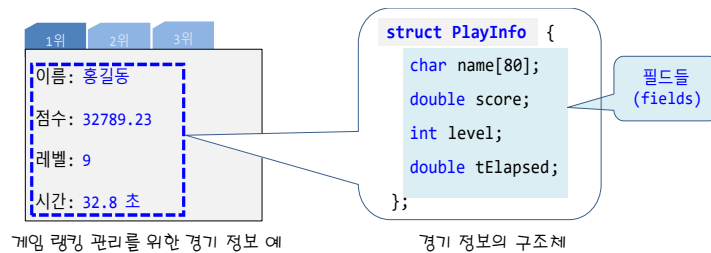
6



구조체란?



- 다른 여러 가지의 다른 유형의 데이터를 묶는 방법
 - 게이머 이름: 문자열로 저장
 - 점수: 정수 또는 실수로 저장
 - 게임 레벨: 정수로 저장
 - 게임 시간: 초단위로 저장하며 실수로 저장

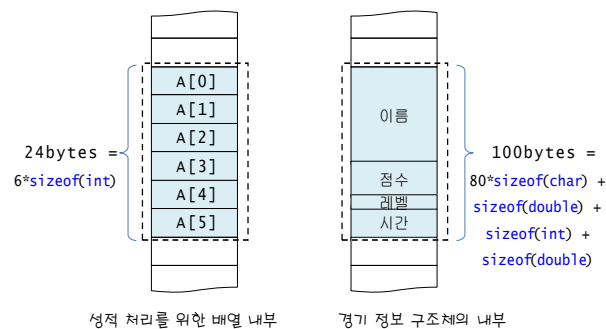


7

배열, 구조체와 파일



- 배열과 구조체 객체는 모두 연속적인 메모리 공간



- 프로그램이 종료되어도 자료를 저장하고 싶다면
 - 파일(file)에 저장해야 함

8

4.2 배열



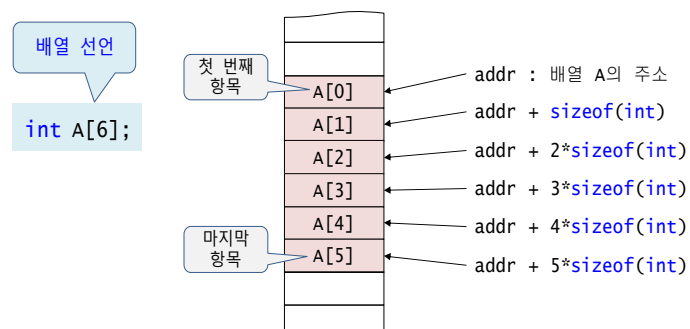
- 배열의 구조
- 배열의 선언과 활용
- Lab: 배열 활용: 근로소득세 계산
- 다차원 배열
- 문자열: 특별한 1차원 배열
- 문자열 처리 함수들

9

배열의 구조



- 자료형 배열이름[배열의_크기];
 - 인덱스는 0부터 배열의_크기 - 1임



10

배열의 선언과 활용



- 배열 선언의 예

```
const int DIM = 10; // 상수 DIM
int size = 20;      // 변수 size를 선언하고 초기화 함
int days[12];       // OK. 12 개의 int값을 저장할 배열 선언
char name[50];      // OK. 50 개의 char값을 저장할 배열 선언
float cost[DIM];    // OK. DIM 개의 float값을 저장할 배열 선언
int score[size];    // 오류! size가 상수가 아니라 변수임
```

- 배열 활용의 예

```
cost[3] = 3.14f; // cost배열의 3번 인덱스의 요소에 3.14f를 복사
name[2] = 'c';   // name배열의 2번 인덱스의 요소에 'c'를 복사
for( int i=0 ; i<7 ; i++ )
    days[i] = 24; // days배열의 모든 요소에 24를 복사
```

11

Lab: 배열 활용: 근로소득세 계산



- 배열을 활용한 근로소득세 계산 함수

```
double calcTax(int income)
{
    const int NumSteps = 5;
    int amount[NumSteps] = { 0, 1200, 4600, 8800, 15000 };
    double tax = 0, rate[NumSteps] = { 0.06, 0.15, 0.24, 0.35, 0.38 };

    for (int i = NumSteps - 1; i>0 ; i--) {
        if (income > amount[i]) {
            tax += rate[i] * (income - amount[i]);
            income = amount[i];
        }
    }
    tax += income*rate[0];
    return tax;
}
```

- 장점은?

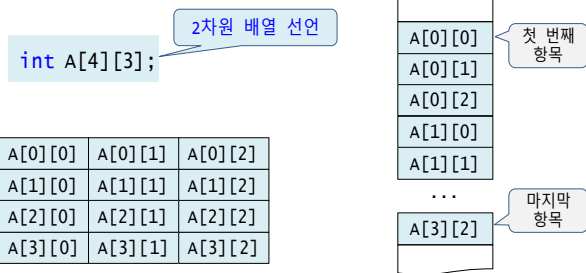
12

다차원 배열



- 자료형 배열이름[행의_크기][열의_크기];

```
int A[4][3];
int B[4][3] = { {1,2,3}, {4,5,6}, {7,8,9}, {10,11,12} };
int C[4][3][5];
```



(a) 항목들의 개념적인 위치

(b) 실제 메모리 안에서의 위치

13

문자열: 특별한 1차원 배열



- char형의 요소를 갖는 1차원 배열

```
char msg[12] = { 'g', 'a', 'm', 'e', ' ', 'o', 'v', 'e', 'r', '\0' };
char s[12] = "game over";
```

	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]	s[11]
s	'g'	'a'	'm'	'e'	' '	'o'	'v'	'e'	'r'	'\0'		

- 문자열의 끝을 나타내는 NULL문자 '\0'
- 문자열 처리 함수들: <string.h> 또는 <cstring>

- 6장에서 string클래스 학습

14

문자열 처리 함수들



- 문자열 처리 함수들: <string.h>, <cstring>
 - strlen(), strcmp(), strcpy(), sprintf(), sscanf() 등

```
#include <string.h>
void main()
{
    char s1[] = "game", s2[20], s3[20];
    strcpy(s2, "over");
    sprintf(s3, "%s %s !", s1, s2);
    printf("strlen(%s) = %d\n", s1, strlen(s1));
    printf("strlen(%s) = %d\n", s2, strlen(s2));
    printf("strlen(%s) = %d\n", s3, strlen(s3));
    printf("strcmp(%s,%s) = %d\n", s1, s2, strcmp(s1,s2));
    printf("strcmp(%s,game) = %d\n", s1, strcmp(s1, "game"));
}
```

```
C:\WINDOWS...
strlen(game) = 4
strlen(over) = 4
strlen(game over !) = 11
strcmp(game,over) = -1
strcmp(game,game) = 0
```

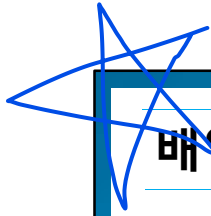
15

4.3 배열과 함수



- 배열과 함수
- 함수에서 배열의 반환?
- 다차원 배열의 매개변수 전달 문제
- Lab. 게임 점수의 내림차순 정렬

16



배열과 함수



- 배열의 이름은 배열의 시작 주소이다.
- 매개 변수로 배열의 길이도 전달해야 한다. ①
 - 문자열의 경우 길이를 전달할 필요가 없다. ②

```
#include <cstring>
int findMaxValue(int a[], int len)
{
    int maxVal = a[0];
    for (int i = 1; i < len; i++)
        if (maxVal < a[i]) maxVal = a[i];
    return maxVal;
}
char findMaxChar(char a[])
{
    char maxChar = a[0];
    for (int i = 1; i < strlen(a); i++)
        if (maxChar < a[i]) maxChar = a[i];
    return maxChar;
}
```

① = int x a

② ? 길이 x

```
void main()
{
    int arr[10] = { 3, 24, 82, 12, 34, 7, 53, 17, 26, 51 };
    char str[] = "game over !";

    printf("max val = %d\n", findMaxValue(arr, 10));
    printf("max char = %c\n", findMaxChar(str));
}
```

max val = 82
max char = v

17

함수에서 배열의 반환?



- 다음 문장은 오류 발생

```
int[] copyArray(const int a[], int len); // 오류
```

- 예) 배열 항목의 역순 배치 함수
 - 두 개의 배열을 인수로 전달해야 함

```
void reverseArray(const int a[], int b[], int len) {
    for (int i = 0; i < len; i++)
        b[len-i-1] = a[i];
}
void reverseString(const char src[], char dst[]) {
    int len = strlen(src);
    for (int i = 0; i < len; i++)
        dst[len - i - 1] = src[i];
    dst[len] = '\0';
}
```

18

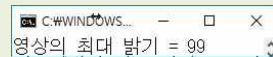
다차원 배열의 매개변수 전달 문제



- 2차원 이상의 배열
 - 매개변수로 보낼 때 반드시 a[][5]와 같이 가로 값을 지정
 - 엉터리 함수

```
int findMaxPixel( int a[][5], int h, int w ) {
    int maxVal = 0;
    for( int i=0 ; i<h ; i++ )
        for( int j=0 ; j<w ; j++ )
            if( maxVal < a[i][j] ) maxVal = a[i][j];
    return maxVal;
}

void main() {
    int img[3][5]= { { 3, 24, 82, 12, 22},
                     { 99, 7, 65, 73, 58},
                     { 20, 7, 9, 48, 29}};
    int maxPixel = findMaxPixel( img, 3, 5 );
    printf( "영상의 최대 밝기 = %d\n", maxPixel );
}
```



19

Lab. 게임 점수의 내림차순 정렬



- 게임을 위한 고득점 랭킹 관리를 위한 프로그램
 - 1등부터 5등까지의 점수를 배열에 저장하려고 함.
 - 무작위 점수 발생 및 배열에 삽입.
 - 배열에는 가장 높은 점수부터 5개만 내림차순으로 정렬
 - 이 과정을 8회 반복.
 - 점수를 추가할 때 마다 전체 배열의 내용을 화면에 출력.



20

```
void addDescending( int a[], int len, int val ) {
    if( val > a[len-1] ) {
        a[len-1] = val;
        for( int i=len-1 ; i>0 ; i-- ) {
            if( val < a[i-1] ) return;
            a[i] = a[i-1];
            a[i-1] = val;
        }
    }
}

void main()
{
    int score[5] = {0,0,0,0,0};
    for( int i=0 ; i<8 ; i++ ) {
        int newScore = rand()%100;
        addDescending( score, 5, newScore );
        printf("[%2d] ", newScore);
        printRanking( score, 5, "순위" );
    }
}
```

Index	Score	Rank
[41]	41	0
[67]	67	1
[34]	67	1
[0]	67	1
[69]	69	2
[24]	69	2
[78]	78	3
[58]	78	3

21

4.4 구조체

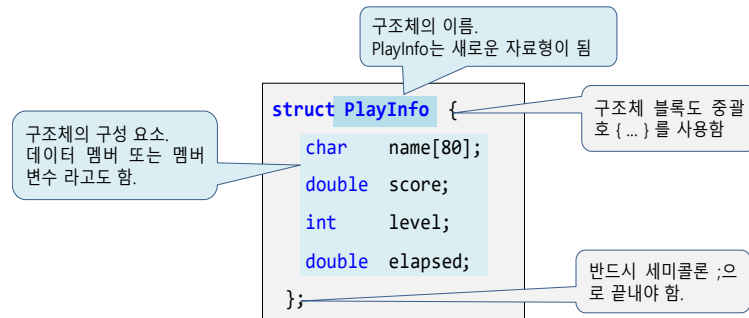
- 구조체란?
- 구조체 (변수) 선언과 활용
- 구조체와 연산자

22

구조체란?



- 다른 여러 가지의 다른 유형의 데이터를 묶는 방법
- 유도 자료형을 만드는 대표적인 방법
- 구조체의 정의



23

구조체 (변수) 선언과 활용



- 구조체 선언

```
struct PlayInfo best;           // C언어 방식의 객체 선언  
PlayInfo challenger;           // C++에서 추가로 허용하는 방법  
  
PlayInfo a, b;                 // 경기 정보 객체 a와 b를 생성함  
PlayInfo list[10];             // 경기 정보 10개를 저장할 배열을 생성함  
PlayInfo challenger = { "Rafael Nadal", 90.0, 9, 30.5 };
```

- 멤버의 선택

– 항목 선택 연산자 .

```
challenger.score = 93.5;  
challenger.level = 9;  
challenger.elapsed = 27.5;  
challenger.name = "Andy Murray"; // 오류: 문자열 복사에는 '='  
strcpy(challenger.name, "Andy Murray"); // OK
```

24

구조체와 연산자



- 대입 연산자만 지원

```
struct Complex {           // 복소수 구조체 정의
    double real;           // 실수부
    double imag;           // 허수부
};
Complex a = {1.0, 2.0};
Complex b = {3.0, 4.0};

b = a;                     // OK: b.real = a.real; b.imag = a.imag;
```

- 다른 연산자: 지원 없음 → 함수를 만들어 사용 함

```
if( a > b )                 // 오류
    printf("a가 b보다 크다.");

bool compare(Complex a, Complex b)
{
    if( a.real > b.real ) return true;
    else return false;
}
```

25

4.5 구조체와 함수



- 구조체와 함수

26

구조체와 함수



```
void printComplex(Complex c, char* msg = "복소수") { ... }
Complex readComplex(char* msg="복소수") { ... }
Complex addComplex(Complex a, Complex b) { ... }
void resetComplex(Complex a) { a.real = a.imag = 0.0; } // ???
void main()
{
    Complex a, b, c;
    a = readComplex("a(입력) = ");
    b = readComplex("b(입력) = ");
    c = addComplex(a, b);
    printComplex(a, " a =");
    printComplex(b, " b =");
    printComplex(c, " a+b =");
    resetComplex(c);
    printComplex(c, "reset(c)="); // ???
}
```

→ 구조, 함수

```
C:\WINDO... - □ ×
a(입력) = 1 2
b(입력) = 3 4
a = 1.00 + 2.00i
b = 3.00 + 4.00i
a+b = 4.00 + 6.00i
reset(c)= 4.00 + 6.00i
```

- resetComplex() 함수가 제대로 동작하지 않음. Why?

27

4.6 값에 의한 호출(call by value)



- 값에 의한 호출

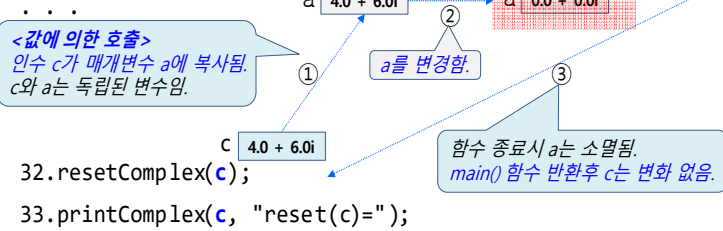
28

값에 의한 호출



- Call-by-value

```
21. void resetComplex(Complex a) { a.real = a.imag = 0.0; }
```



29

4.7. 파일 입출력



- 텍스트 파일과 이진 파일
- 파일 입출력 함수
- 텍스트 파일 입출력 예
- 이진 파일 입출력 예

30

텍스트 파일과 이진 파일



- 텍스트 파일
 - 자료들을 모두 문자열로 변환하여 저장
 - 대부분의 문서 편집기에서 내용을 읽을 수 있음
- 이진 파일
 - 자료들이 메모리에 있는 그대로를 저장
 - 보통의 문서 편집기에서 그 내용을 알 수 없음

- 파일 열기

```
FILE * fopen ( const char * filename, const char * mode);
```

- 파일 닫기

```
int fclose ( FILE* fp );
```

31

파일 입출력 함수



- 텍스트 모드의 파일 입출력 함수

```
int fprintf( FILE*, const char *, ... );  
int fscanf( FILE*, const char *, ... );
```

– 사용 예

```
fprintf(stdout, "값 = %d", 10); // printf("값 = %d", 10);와 동일  
fscanf(stdin, "%d", &val); // scanf("%d", &val);와 동일
```

- 이진 모드의 파일 입출력 함수

```
size_t fwrite(void* ptr, size_t size, size_t count, FILE* fp );  
size_t fread (void* ptr, size_t size, size_t count, FILE* fp );
```

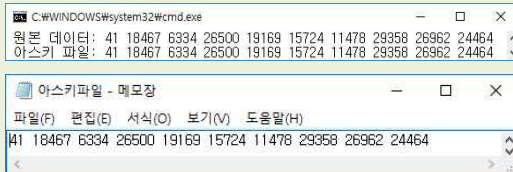
32

텍스트 파일 입출력 예



```
void fprintfArray(int a[], int len, FILE* fp = stdout)
{
    for (int i = 0; i < len; i++)
        fprintf(fp, "%d ", a[i]);
    fprintf(fp, "\n");
}

void main()
{
    FILE *fp;
    ...
    fprintfArray(a, 10);
    fp = fopen("아스키파일.txt", "w");
    if (fp != NULL) {
        fprintfArray(a, 10, fp);
        fclose(fp);
    }
    ...
}
```

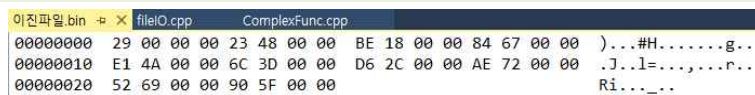


33

이진 파일 입출력 예



```
...
fp = fopen("이진파일.bin", "w");
if (fp != NULL) {
    fwrite(a, sizeof(int), 10, fp);
    fclose(fp);
}
fp = fopen("이진파일.bin", "r");
if (fp != NULL) {
    fread(b, sizeof(int), 10, fp);
    printf("이진 파일: ");
    fprintfArray(b, 10);
    fclose(fp);
}
```



34

4.8 응용: 랭킹 보드 프로그램



- 구조체와 처리 함수 (4x4 퍼즐 게임용)
- 파일 저장 형식
- 데이터 처리 과정

35

구조체와 처리 함수, 저장 형식



- 구조체와 처리 함수 (4x4 퍼즐 게임용)

```
struct PlayInfo {
    char name[80];    // 선수 이름
    int nMove;        // 퍼즐을 움직인 횟수
    double tElapsed;  // 경기 소요 시간
};

void readRanking(char *filename): 랭킹을 파일에서 읽음
void storeRanking(char *filename): 현재 랭킹을 파일로 저장
void printRanking(): 현재 랭킹을 화면에 출력
void addRanking(int nmove, double elapsed):
```



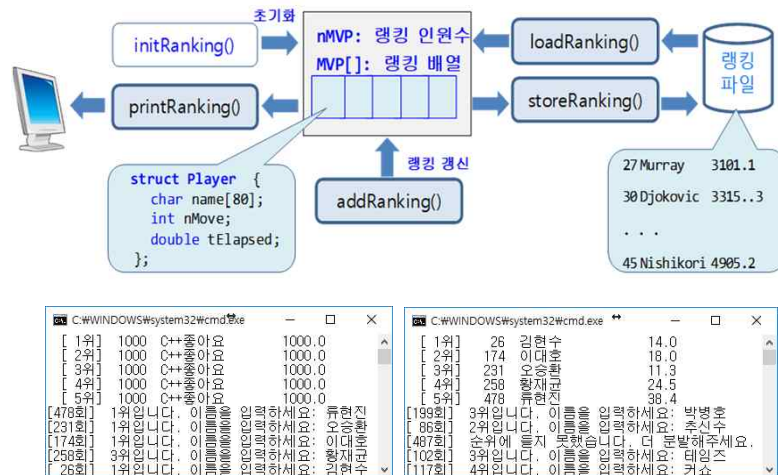
- 파일 저장 형식

ranking - 메모장

파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)
184	이준수	40.3		
216	김민준	39.0		
303	김민준	15.2		
308	김민준	5.3		
347	김민준	94.8		

36

데이터 처리 과정

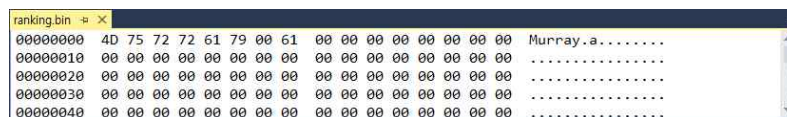


37

• 텍스트 모드 저장



• 이진 모드 저장



38

4.9. 응용: 4x4 퍼즐 게임



- 4x4 퍼즐 게임이란?
- 게임 처리 과정
- 구현
- 실행 결과

39

4x4 퍼즐 게임이란?

12	15	6	10
4	9	5	8
14	13		2
1	7	11	3

- 구조체와 처리 함수 (4x4 퍼즐 게임용)
 - 최초에는 퍼즐이 모두 제 위치에 있어야 함
 - 엔터를 누르면 퍼즐 조각을 섞는다. (난수 발생)
 - 퍼즐이 섞이는 과정을 화면에서 볼 수 있도록 출력에 시간 지연
 - 힌트: <windows.h>에 선언되어 있는 Sleep() 함수 사용
 - 퍼즐이 섞이면 엔터를 눌러 게임을 시작
 - 퍼즐 조각의 이동에는 화살표 키 사용
 - 키를 누르면 바로 반응하도록 함.
 - 모든 퍼즐 조각들이 제자리를 찾으면 게임이 종료
 - 퍼즐 조각을 움직인 전체 횟수, 걸린 시간을 계산
 - 이동 횟수를 바탕으로 랭킹을 갱신
 - 현재 랭킹을 화면에 출력
 - 랭킹은 파일에 저장되어야 함.

40

게임 처리 과정



41

구현



```
enum Direction { Left = 75, Right = 77, Up = 72, Down = 80 };

static int DirKey[4] = { Left, Right, Up, Down };
static int map[DIM][DIM]; // 퍼즐 맵
static int x, y;          // 빈 칸의 위치 (행:y, 열:x)
static int nMove;         // 이동한 횟수
static clock_t tStart;    // 기준 시각

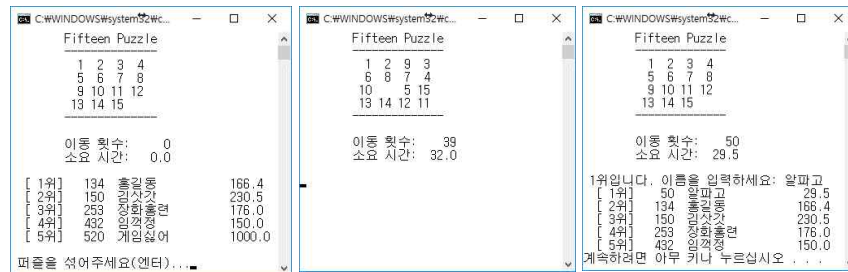
static void init() {...}
static void display() {...}
static bool move(int dir) {...}
static void shuffle(int nSh){...}
static bool isDone() {...}
static int getDirKey() {
    return getche() == 224 ? getche() : 0;
}

int playFifteenPuzzle() { ... }
```

```
#include "Ranking.h"
extern int playFifteenPuzzle();
void main() {
    loadRanking("ranking.txt");
    int rank = playFifteenPuzzle();
    printRanking();
    storeRanking("ranking.txt");
}
```

42

실행 결과



43

4장 요약문제, 연습문제, 실습문제



44



감사합니다!