

Data Structure Assignment

2016320205 이지혜

2015410072 김재현

2014250413 이상화

1. 학생들의 성적을 처리하여 평균과 표준편차를 구하는 프로그램 / 이지혜

```
#include <stdio.h>    표준 입출력을 위한 헤더
#include <stdlib.h>    배열 요소개수를 세는 _countof 함수를 사용하기 위한 헤더
#include <time.h>      랜덤의 시드를 만들기 위한 시간 헤더
#include <math.h>      식의 계산의 위한 수학 헤더
```

```
typedef struct student {...} Student;
```

struct student 라는 자료형을 typedef를 사용하여 Student라는 자료형으로 선언하였다.
이름과 학번, 성적을 각각 char 배열과 int형으로 저장한다.

```
void init_student (Student s[]);
```

입력받은 Student 배열을 초기화한다. 학번과 이름은 Student_info.txt라는 텍스트 파일로부터 입력받으며, 파일을 읽
어오지 못하면 망함을 출력하고 return한다. 이를 확인하기위해 입력받은 값을 출력한다.
또한 성적을 0~100 사이 중에 임의의 값으로 저장하고, 이를 출력한다.

```
double get_mean (Student s[], int std_num);
```

Student 배열과 학생 수를 입력받아 Student 구조체의 score를 읽어와 평균을 계산하고, 그 값을 return한다.

```
double get_square_mean (Student s[], int std_num);
```

Student 배열과 학생 수를 입력받아 Student 구조체의 score를 읽어와 제곱 평균을 계산하고, 그 값을 return한다.

```
double get_deviation (Student s[], int std_num);
```

Student 배열과 학생 수를 입력받아 get_mean 함수와 get_square_mean 함수를 호출하여 평균값과 제곱평균값을 얻
고, 이를 이용해 표준편차를 계산해 그 값을 return한다.

```
int main(void) {...}
```

Student 배열을 선언한 뒤 init_student 함수로 배열을 초기화한다. 또한 get_mean, get_deviation 함수를 호출해 평
균과 표준편차를 구하고, 학생들의 성적과 표준편차를 출력한다.

2. 재귀 함수를 사용하여 문자를 역순으로 출력하는 프로그램 / 이상화

```
#include <stdio.h>    표준 입출력을 위한 헤더
```

```
void reverse(char *s);
```

역순으로 출력하는 재귀호출함수로, 문자열의 끝, 0을 만나면 빠져나가고, 그것이 아닐 경우에는 포인터를 1더해서
다시 재귀 호출을 한다. 그리고 그 과정에서 나오는 현재 위치의 문자를 출력하는 함수이다.

```
int main(void) {...}
```

문자열을 입력하고, 역순 출력 재귀함수를 호출하여 그 값을 출력하는 메인함수이다.

3. 주어진 2차원 배열의 영역 안의 한 점을 시작으로, 안쪽을 채우는 재귀함수 프로그램 / 김재현

```
#include <stdio.h>    표준 입출력을 위한 헤더
```

```
enum { LFUP, LFDN, ...};
```

방향을 쉽게 표기할 수 있도록 선언된 열거형이다.

```
struct Arr {...};
```

배열의 정보가 담긴 구조체로, 배열에 담긴 내용과 배열의 행·열 끝을 담고 있다.

10x10 크기의 2차원 배열이 입력으로 들어온다고 가정하였으며, 끝의 NULL을 고려하여 조금 더 크게 할당한다.

```
void full_one(struct Arr *arr, int row, int column, int direction)
```

배열의 상하좌우를 채우는 함수이다.

인자로는 0을 1로 바꿀 배열 구조체의 포인터, 현재 가리키고 있는 행·열, 가리키고 있는 방향을 받는다.

먼저 현재 가리키고 있는 행 또는 열이 배열의 범위를 벗어날 경우 return한다.

그리고 현재 가리키고 있는 행 또는 열이 배열의 끝인 경우 칸을 바꾼 뒤 return한다.

그렇지 않은 경우에는 재귀함수를 이용하여 현재 탐색하고 있는 방향(direction)으로 한 번 더 탐색한다.

탐색의 효율성을 위해(겹치는 부분을 탐색하지 않도록) 맨 처음에는 방향을 -1로 받아 시작점의 상하좌우를 탐색한다.

```
void find_dir(struct Arr *arr, int row, int column, int direction)
```

대각선으로 다음 탐색할 방향을 지정하는 함수이다.

인자로는 0을 1로 바꿀 배열 구조체의 포인터, 현재 가리키고 있는 행·열, 가리키고 있는 방향을 받는다.

먼저 현재 가리키고 있는 행 또는 열이 배열의 범위를 벗어날 경우 return한다.

그렇지 않은 경우에 현재 칸이 0이라면 1로 바꾸고, 대각선 방향에 따라 full_one 함수를 이용하여 상하좌우를 탐색한다. 상하좌우의 탐색이 끝나면 현재 가리키고 있는 행 또는 열이 배열의 끝인지 확인하고, 그렇다면 return한다.

그렇지 않은 경우에는 재귀함수를 이용하여 현재 탐색하고 있는 대각선 방향으로 한 번 더 탐색한다.

```
int main(void) {...}
```

struct Arr {...}; 을 선언한 뒤 주어진 배열로 내용을 초기화한다. 각 행과 열의 끝을 지정한 뒤, 과제에 시작점의 좌표를 지정한다. 맨 처음 full_one 함수의 direction 인자를 -1로 주어 시작점과, 시작점의 상하좌우를 탐색한 뒤 좌상, 좌하, 우상, 우하를 차례대로 탐색한다.