



퀴즈 시스템 개발

- 이번 장에서는 문제들이 저장된 파일을 만들고, 파일에서 랜덤하게 정해진 개수의 문제를 선택하여 사용자에게 제시하고 사용자의 답변을 채점하는 퀴즈 시스템을 만들어보자.
- 주제는 “C언어 프로그래밍”이다.

Welcome to the C Programming Quiz!

Question 1:

다음 코드의 실행 결과는 무엇입니까? #include <stdio.h> int main(void) { int x = 5; printf("%d\n", x << 1); return 0; } A) 10 B) 2 C) 20 D) 6

Your answer: A

잘못된 답변입니다. 다음 기회에

...

...

...

Question 5:

다음 코드의 실행 결과는 무엇입니까? #include <stdio.h> int main(void) { int x = 5; int *ptr = &x; printf("%p\n", ptr); return 0; } A) 변수 x의 주소 B) 5 C) 컴파일 오류 D) 정의되지 않는다.

Your answer: A

맞았습니다. 5점을 획득하였습니다.

전체 점수: 15

감사합니다!



구현 기능

- 사지선다 문제들이 암호화된 파일 `questions.bin`에 저장된다.
- 암호화는 **XOR** 암호 기법을 사용한다.
- 각 문제는 정답과 점수를 가지고 있다.
- 사용자의 답변은 문제의 정답과 비교된다. 맞았는지, 틀렸는지를 화면에 즉시 출력한다.
- 사용자의 총점을 마지막에 출력한다.



문제 파일의 구조

- 각 문제는 다음과 같은 세 부분으로 이루어진다.
 - 질문지와 선택지
 - 정답(A, B, C, D 중 하나)
 - 부여 점수

questions.txt

다음 코드의 실행 결과는 무엇입니까? #include <stdio.h> int main(void) { int x = 5; printf("%d\n", x++); return ; } A) 5 B) 6 C) 4 D) 0

A

정답

질문지와 선택지

5

다음 코드의 실행 결과는 무엇입니까? #include <stdio.h> int main(void) { int arr[] = {1, 2, 3, 4, 5}; printf("%d\n", *(arr + 3)); return 0; } A) 3 B) 4 C) 2 D) 1

B

부여 점수

5

...



[R U# 암호화

- XOR 암호화(Exclusive OR Encryption)는 비트 단위로 데이터를 암호화하는 간단한 암호화기술 중 하나이다. 이 기술은 원본 데이터와 암호 키 간의 XOR(배타적 논리합) 연산을 사용하여 암호화를 수행한다.

$$\begin{array}{r|l} & 00110101 \text{ (평문)} \\ \text{XOR} & 11100011 \text{ (키)} \\ \hline & 11010110 \text{ (암호문)} \end{array}$$

$$\begin{array}{r|l} & 11010110 \text{ (암호문)} \\ \text{XOR} & 11100011 \text{ (키)} \\ \hline & 00110101 \text{ (평문)} \end{array}$$



암호화된 파일 해독하기

```
#include <stdio.h>
// XOR 암호 해제 함수
void decrypt_file(const char* input_filename, const char* output_filename) {
    // 입력 파일 및 출력 파일 열기
    FILE* input_file = fopen(input_filename, "rb");
    FILE* output_file = fopen(output_filename, "wb");

    // 파일 열기 오류 처리
    if (input_file == NULL || output_file == NULL) {
        printf("파일 오픈 오류!\n");
        return;
    }

    int ch;
    // 파일의 끝까지 반복하여 각 바이트에 XOR 연산 수행
    while ((ch = fgetc(input_file)) != EOF) {
        // XOR 연산을 통해 암호 해제된 데이터를 출력 파일에 쓰기
        fputc(ch ^ 0xFF, output_file); // XOR 암호 해제
    }
}
```



암호화된 파일 해독하기

```
//파일 닫기
fclose(input_file);
fclose(output_file);
}

int main(void) {
    // 암호 해제된 파일 생성
    decrypt_file("questions.bin", "questions.txt");
    return 0;
}
```



문제들이 저장되는 구조체 배열

- 문제들은 프로그램에서 어디에 저장될까? 바로 구조체 배열이다. 구조체 배열은 퀴즈 시스템에서 각 문제의 정보를 저장하는 데 사용된다. 이 구조체 배열은 프로그램이 실행될 때 생성되며, 배열의 각 요소는 하나의 문제를 나타낸다.

```
#define MAX_QUESTION_LENGTH 200
#define MAX_ANSWER_LENGTH 50
#define MAX_QUESTIONS 20

// 퀴즈 구조체 정의
typedef struct {
    char question[MAX_QUESTION_LENGTH];
    char answer[MAX_ANSWER_LENGTH];
    int score;
} Quiz;

Quiz quizzes[MAX_QUESTIONS]; // 구조체 배열 정의
```



VWHS#5 해독된 문제 파일 읽기

```
// 퀴즈 구조체 정의
typedef struct {
    char question[MAX_QUESTION_LENGTH];
    char answer[MAX_ANSWER_LENGTH];
    int score;
} Quiz;

// 퀴즈 파일에서 퀴즈를 읽어오는 함수
int read_quiz_file(const char* filename, Quiz* quizzes, int max_quizzes) {
    FILE* file = fopen(filename, "r");
    if (file == NULL) {
        printf("파일 오픈 오류!\n");
        return 0;
    }

    int num_quizzes = 0;
    while (num_quizzes < max_quizzes && fscanf(file, "%[^\\n]\\n%[^\\n]\\n%d\\n", quizzes[num_quizzes].question, quizzes[num_quizzes].answer, &quizzes[num_quizzes].score) == 3) {
        num_quizzes++;
    }

    fclose(file);
    return num_quizzes;
}
```




VWHS#5 해독된 문제 파일 읽기

```
int main(void) {  
    Quiz quizzes[MAX_QUESTIONS];  
  
    // 암호 해제된 파일에서 퀴즈 읽어오기  
    num_quizzes = read_quiz_file("questions.txt", quizzes, MAX_  
QUESTIONS);  
}
```



scanf의 막강한 기능

```
int num_quizzes = 0;
while (num_quizzes < max_quizzes && fscanf(file, "%[^\\n]\\n%[^\\n]\\n%d\\n", quizzes[num_quizzes].question, quizzes[num_quizzes].answer, &quizzes[num_quizzes].score) == 3) {
    num_quizzes++;
}
```

- `%[^\\n]\\n`: 줄 바꿈 문자가 나오기 전까지의 문자열을 읽어온다. 이는 퀴즈의 질문을 나타낸다.
- `%[^\\n]\\n`: 줄 바꿈 문자가 나오기 전까지의 문자열을 읽어온다. 이는 퀴즈의 정답을 나타낸다.
- `%d\\n`: 정수를 읽어온다. 이는 퀴즈의 점수를 나타낸다.

```
질문 1: 다음 코드의 실행 결과는 무엇입니까? #include <stdio.h> int main() { ... } A)
5 B) 6 C) 4 D) 0
A
5
```

```
\\n
\\n
\\n
```



VWHS#6 전체 문제에서 랜덤하게 선택하기

```
// 랜덤하게 퀴즈를 선택하는 함수
int select_random_quizzes(int num_quizzes, int num_select, int* selected)
{
    if (num_select > num_quizzes) {
        printf("선택된 문제의 수가 전체 문제보다 많습니다!\n");
        return 0;
    }
    // 선택된 문제를 나타내는 배열 초기화
    int selected_flag[MAX_QUESTIONS] = { 0 };

    // 랜덤하게 퀴즈 선택
    for (int i = 0; i < num_select; i++) {
        int index = rand() % num_quizzes; // 랜덤 인덱스 생성
        // 중복 확인
        while (selected_flag[index]) {
            index = rand() % num_quizzes;
        }
        // 중복되지 않은 인덱스 선택
        selected[i] = index;
        selected_flag[index] = 1;
    }
    return 1;
}
```



VWHS#6 7 퀴즈 진행하기

```
// 퀴즈 시험을 진행하는 함수
void take_quiz(Quiz* quizzes, int* selected, int num_select) {
    char answer[MAX_ANSWER_LENGTH];
    int total_score = 0;

    printf("Welcome to the C Programming Quiz!\n");

    for (int i = 0; i < num_select; i++) {
        printf("\nQuestion %d:\n", i + 1);
        printf("%s\n", quizzes[selected[i]].question);
        printf("Your answer: ");
        scanf("%s", answer);

        if (strcmp(answer, quizzes[selected[i]].answer) == 0) {
            printf("맞았습니다. %d점을 획득하였습니다. \n",
quizzes[selected[i]].score);
            total_score += quizzes[selected[i]].score;
        }
        else {
            printf("잘못된 답변입니다. 다음 기회에 \n");
        }
    }

    printf("\n전체 점수: %d\n", total_score);
    printf("감사합니다!\n");
}
```



함수 작성

```
int main(void) {
    int num_quizzes, num_select;
    int selected[5];
    Quiz quizzes[MAX_QUESTIONS];

    srand(time(NULL)); // 시드 설정

    // 암호 해제된 파일 생성
    decrypt_file("questions.bin", "questions.txt");

    // 암호 해제된 파일에서 퀴즈 읽어오기
    num_quizzes = read_quiz_file("questions.txt", quizzes, MAX_QUESTIONS);

    // 랜덤하게 퀴즈 선택
    num_select = 5; // 5개의 퀴즈 선택
    if (select_random_quizzes(num_quizzes, num_select, selected)) {
        // 퀴즈 시험 진행
        take_quiz(quizzes, selected, num_select);
    }
    return 0;
}
```



도전 문제

- 사용자가 문제를 좀 더 알기 쉽게 볼 수 있도록 문제에 포함된 코드를 다음과 같이 표시해보자.

Welcome to the C Programming Quiz!

Question 1:

다음 코드의 실행 결과는 무엇입니까?

```
#include <stdio.h>
int main(void) {
    int x = 5;
    printf("%d\\n", x << 1);
    return 0;
}
```

A) 10 B) 2 C) 20 D) 6

Your answer: A

잘못된 답변입니다. 다음 기회에