

c프로그래밍 및 실습

프로젝트 제목

최종 보고서

제출일자: 2023.12.24

제출자명: 김진하

제출자 학번: 221473

1. 프로젝트 목표

번역기를 통한 영어가 아닌 실제 생활에서 많이 쓰이는 영어 표현을 배우며 고객의 생활 영어 향상을 목표로 한다.

1) 배경 및 필요성

영어 회화는 어디에서나 중요하게 요구될 수 있는 역량 중 하나다. 하지만 단순히 번역기로 잘못 배운 영어회화는 상황에 맞지 않는 표현을 쓸 확률이 높아 실수하게 될 수도 있고, 누군가는 오해를 할 수도 있다. 이런 상황을 해결하기 위해 쉽게 접할 수 있는 생활영어 회화 배우기 프로그램을 만들었다.

2) 프로젝트 목표

사람들이 자주 쓰는 어색한 표현을 올바른 표현 및 상황에 맞는 표현법을 알려주며 고객의 실전 생활영어의 향상을 목표로 한다.

3) 차별점

기존 번역기 같은 경우 한국어로 문장을 입력하면 바로 영어로 번역되지만 단순히 한 문장만 보여주어 그 상황에 맞지 않는 표현을 쓸 수도 있고, 어색한 표현일 수도 있다. 이 문제를 해결하기 위해 상황에 적합한 표현과 어색한 표현 등등을 보여주며 자신의 언어 실수를 쉽게 이해하며 바로잡을 수 있고, 또한 그 나라의 문화를 이해할 수도 있다는 차별점이 있다

2. 기능 계획

1) 영어 회화 배우기 챗터

- 상황에 따른 영어 회화 수집

Ex) 인사하기, 물어보기 등등

- 상황에 따른 영어회화 별 세분화

Ex)1. 인사하기

1) 인사말 2) 이름 묻기

2) 단어 배우기 챗터

- 자주 실수를 범하는 단어표현 보여주기

Ex) 1. 약속 2. 가다오다 등등

-잘못된 표현의 단어를 정정하며 올바른 표현 출력 및 설명

Ex) promise -> plan

구조체를 사용하여 영어회화 및 단어의 데이터를 저장할 예정

Void printConversation: 대화 출력함수

Void PrintWord: 단어 출력함수

3. 기능 구현

(1) study_eng.h 헤더파일 안에서 대화 및 단어 데이터 구조체선언 & 함수 선언

- 입출력

situatin: 영어 회화배우기 (상황) 문자열배열 정의

Expression: 영어 회화배우기 (표현) 문자열 배열 정의

Correction: 영어 회화 배우기(정정) 문자열 배열 정의

Word는 단어를 저장하는 배열, explanation은 해당 단어의 설명을 저장하는 배열

- 설명

이 구조체는 대화 데이터를 저장하기 위한 것으로, 세가지 멤버변수인 situation, expression, correction을 가지고 있다. 각각은 문자열을 저장할 수 있는 고정된 크기의 배열로 선언되어 있다. 데이터를 저장하고 출력 및 저장을 위해 구조체를 선언하였다.

- 적용된 배운 내용

: 구조체 선언 및 함수 선언

```
// 대화 데이터 구조체 정의
struct Conversation {
    char situation[50];
    char expression[100];
    char correction[100];
};
```

```
// 단어 데이터 구조체 정의
struct Word {
    char word[50];
    char explanation[100];
};
```

--함수 선언 --

saveConversations: 대화 데이터를 저장하는 함수, Conversation 구조체의 배열과 배열의 크기를 인자로 받는다.

Void saveWords: 단어 데이터를 저장하는 함수, word 구조체의 배열과 배열의 크기를 인자로 받는다.

Void PrintConversation: 대화 데이터를 출력하는 함수로, Conversation 구조체의 포인터를 받는다.

Void printWord: 단어 데이터를 출력하는 함수로, Word 구조체의 포인터를 받는다.

배운 내용: 함수선언, 포인터 적용

```
// 함수 선언
void saveConversations(struct Conversation *conversations, int size);
void saveWords(struct Word *words, int size);
void printConversation(struct Conversation *convo);
void printWord(struct Word *wd);
```

(2) study_eng.c 헤더파일에 정의된 구조체와 함수들을 구현한 코드

출력:

printConversation 함수: 이 함수는 struct Conversation 포인터를 매개변수로 받는다.

Convo-> situation은 구조체 내의 상황(situation)문자열을 출력한다. 즉 각각의 정보는 특정 대화 상황에 대한 정보를 출력하는 데 사용된다.

printWord 함수: 이 함수는 struct Word 포인터를 매개변수로 받는다.

Wd-> word는 구조체 내의 단어(word)문자열을 출력한다.

설명: 두개의 함수를 정의하여 구조체를 인자로 받아 해당하는 정보를 화면에 출력하는 역할을 한다.

배운 내용: 함수구현, 구조체의 포인터 전달하여 호출하기, 포인터적용

```
#include "study_eng.h"

#include <stdio.h>

// 대화 출력 함수
void printConversation(struct Conversation *convo) {
    printf("---%s---\n", convo->situation);
    printf("%s\n", convo->expression);
    printf("----정정----\n");
    printf("%s\n", convo->correction);
}

// 단어 출력 함수
void printWord(struct Word *wd) {
    printf("---%s---\n", wd->word);
    printf("%s\n", wd->explanation);
}
```

-메인 함수 내에서 구현된 구조체 배열

설명:

stuct Conversation conversations[]:

이 배열은 각 대화 정보를 담고 있어 상황(situation), 표현(expression), 정정(correction)으로 구성된다.

Struct Word words[]:

이 배열은 단어 정보를 담고 있어 단어(word)와 그에 해당하는 설명(explation)으로 구성된다.

배운 내용: 구조체 구현

// 대화 데이터 초기화

```
struct Conversation conversations[] = {
    {"인사말", "hey",
     "hey: 친해진 뒤 사용! Wnhow are you: 전에 만난 적 있는 사람에게 "
     "사용!"},
    {"이름묻기", "hi my name is jina", "hi i'm jina"},
    {"오랜만에 만날때", "nice to meet you", "good to see you again"},
    {"첫 만남 무례한 질문", "how old are youWnWhat's your job",
     "정말 물어보고 싶을때->WnMay i ask you age?Wn what do you do for a "
     "living?"},
    {"친해지기 위해 쓰는 표현", " how old are you ",
     "what do you do in your free time"},
    {"맛이 어떤지 물어보기", "is it delicious?", "how does it taste?"},
    {"주문시 시간이 필요할때", " wait a minute",
     "atually, i think we need anoter time"},
    {"메뉴 결정을 했을때", "I decided", "I made a decision"},
    {"서비스인지 물어보기", "it's service?",
     "it's on us/ it's on the house"}};
```

// 단어 데이터 초기화

```
struct Word words[] = {
    {"가다 오다"},
    {"come: 화자와 청자가 가까워질 때 사용Wngo: 반대로 멀어질 때 사용"},
    {"패딩-padding", "puffer jacket"},
    {"우리 동네-my neighbor", "my neighborhood"},
    {"버스 정류장-bus station", "bus stop"},
    {"후드티-hood-T", "hoodie"},
    {"킥보드-kick board", "kick scooter"},
    {"미혼- solo", "single"},
    {"미팅-meeting", "blind date"},
    {"요리사-cooker", "cook"},
    {"토스트-toast", "grilled sandwich"},
    {"화장실-toilet", " bathroom "},
    {"아파트-apart", "apartment/building/complex"},
    {"원룸-one room", "studio apartment"},
    {"콘도-condo", "hotel"},
    {"빌라-villa/mansion", "apartment"},
    {"한시간 - a hour ", "an hour"},
    {"이틀전에-two days before", " two days ago "},
    {"10분 후에-after 10 minutes", "in 10 minutes"},
    {"그냥-just", "just because"},
    {"한국사람들-Korean people", "Koreans"}};
```

대화 데이터 및 단어데이터를 파일에 저장하는 함수 구현

출력:

fprintf: 배열에 있는 데이터를 순회하며 파일에 해당 데이터를 쓰기 위한 함수

for루프를 사용하여 배열에 있는 각 단어 정보를 파일에 기록한다.

설명:

배열에 있는 데이터를 순회하며 파일에 해당 데이터를 쓰기 위해 fprintf 함수를 사용한다. 파일을 성공적으로 열고 닫는지 확인하여 파일 작업을 수행한다.

위 함수들은 대화 데이터 및 단어 데이터를 words.txt파일에 저장한다.

세부 코드 설명:

Fopen_s(&file, "words.txt", "w")

: words.txt라는 파일을 쓰기 모드("w")로 연다. 파일을 열지 못할 경우 (NULL일 경우)오류 메시지를 출력하고 함수가 종료된다.

Fclose(file): 파일을 닫는다.

배운 내용: txt파일 저장, for문, 포인터 적용, 조건문, 파일 입출력


```

// 대화 데이터를 파일에 저장하는 함수
void saveConversations(struct Conversation *conversations, int size) {
    FILE *file;
    fopen_s(&file, "conversations.txt", "w"); // 대화 데이터를 저장할 파일명
    if (file == NULL) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    for (int i = 0; i < size; ++i) {
        fprintf(file, "%s\n%s\n%s\n", conversations[i].situation,
            conversations[i].expression, conversations[i].correction);
    }

    fclose(file);
}

// 단어 데이터를 파일에 저장하는 함수
void saveWords(struct Word *words, int size) {
    FILE *file;
    fopen_s(&file, "words.txt", "w"); // 단어 데이터를 저장할 파일명
    if (file == NULL) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    for (int i = 0; i < size; ++i) {
        fprintf(file, "%s\n%s\n", words[i].word, words[i].explanation);
    }

    fclose(file);
}

```

(3) 메인함수 구현

-포인터를 사용해 구조체 초기화

Conversations 및 words는 각각 대화와 단어를 가리키는 구조체 포인터이다. 동적할당으로 메모리를 할당하고 데이터를 저장할 공간을 마련하는 용도로 선언되었다.

배운내용: 구조체, 포인터

```
#include <stdio.h>
#include "study_eng.h" // 구조체 및 함수 선언이 정의된 헤더 파일

// 대화 데이터 초기화 (포인터로 변경)
struct Conversation *conversations;

// 단어 데이터 초기화 (포인터로 변경)
struct Word *words;
```

- 대화와 단어 배열에 대한 메모리 동적 할당

Malloc()함수를 사용하여 conversations와 words가 가리키는 메모리 영역에 대화와 단어를 저장할 공간을 할당한다. 대화 데이터에는 9개의 대화가, 단어 데이터에는 20개의 단어가 저장될 수 있는 공간을 할당한다.

배운 내용: 동적 메모리 할당

```
// main함수
int main() {
    // 대화와 단어 배열에 대한 메모리 동적 할당
    conversation = malloc(sizeof(struct Conversation) * 9);
    words = malloc(sizeof(struct Word) * 20);

    // 메모리 할당이 제대로 이루어졌는지 확인
    if (conversations == NULL || words == NULL) {
        printf("메모리 할당에 실패했습니다.\n");
        return 1; // 프로그램 종료, 오류 코드 반환
    }
}
```

생활 영어 회화 배우기 챕터

입력:

Choice, Choicee, Choic, cho: 이 변수들은 사용자가 메뉴를 선택한 후 서브메뉴의 선택을 저장하는 데 사용된다.

출력:

Printf를 통해 콘솔에 문장이 출력된다. Ex) ----쉽게 배우는 생활영어----

설명:

메뉴를 통해 다양한 선택지를 보고 원하는 번호를 선택하여 그에 따라 대화함수가 호출되고 그 결과를 콘솔에 출력된다.

코드 세부 설명:

Do-while루프가 시작되면서 3번을 누를 시 빠져나간다.

switch-case문에 따라 사용자의 선택이 구별된다.

If-else문을 사용하여 다른 옵션을 제공하고, 그에 맞는 함수를 호출하여 대화가 출력된다.

배운 내용: do-while반복문, (if, switch)조건문, 함수,

```

int choice = 0;
int choicee = 0;
int choic = 0;
int cho = 0;
printf("----쉽게 배우는 생활영어----\n\n");
do {
    printf("1) 회화 배우기   2) 단어 배우기   3) 종료하기\n\n");
    scanf_s("%d", &choice);

    switch (choice) {
        // 생활영어 회화 배우기 챕터
        case 1: {
            int subChoice = 0;
            do {
                printf("\n <<생활 영어 회화 배우기>>   \n");
                printf("1) 인사하기  2) 물어보기  3) 주문하기  4) 종료하기\n\n");
                printf("선택: ");
                scanf_s("%d", &subChoice);

                if (subChoice == 1) {
                    printf(" ---인사하기--- \n");
                    printf("1) 인사말\n 2) 이름묻기\n 3) 오랜만에 만날 때\n");
                    scanf_s("%d", &choicee);

                    if (choicee == 1) {
                        printConversation(&conversations[choicee - 1]);
                    } else if (choicee == 2) {
                        printConversation(&conversations[choicee - 1]);
                    }
                }
            } while (subChoice != 4);
        }
    }
} while (choice != 3);

```

```

    if (choicee == 1) {
        printConversation(&conversations[choicee - 1]);
    } else if (choicee == 2) {
        printConversation(&conversations[choicee - 1]);
    } else {
        printConversation(&conversations[choicee - 1]);
    }
} else if (subChoice == 2) {
    printf("---물어보기---\n\n");
    printf(
        "1) 첫 만남 무례한 질문\n2) 친해지기 위해서 쓰는 표현\n3) "
        "맞이 어떤지 물어보기\n");
    scanf_s("%d", &choic);

    if (choic == 1) {
        printConversation(&conversations[4 - choic]);
    } else if (choic == 2) {
        printConversation(&conversations[6 - choic]);
    } else {
        printConversation(&conversations[8 - choic]);
    }
} else if (subChoice == 3) {
    printf("---주문하기---\n\n");
    printf(
        "1) 주문시 시간이 필요할때\n2) 결정을 했을때\n3) 서비스인지 "
        "물어보기");
    scanf_s("%d", &cho);

    if (cho == 1) {
        printConversation(&conversations[7 - cho]);
    } else if (cho == 2) {

```

```

        if (cho == 1) {
            printConversation(&conversations[7 - cho]);
        } else if (cho == 2) {
            printConversation(&conversations[9 - cho]);
        } else if (cho == 3) {
            printConversation(&conversations[11 - cho]);
        } else {
            printf("잘못된 선택입니다.");
            break;
        }
    } else if (subChoice != 4) {
        printf("잘못된 선택입니다. 다시 선택하세요.\n");
    }
} while (subChoice != 4);
break;
}

```

-1 생활영어 (단어 배우기) 챗터

입력:

Subchoice,cho : 서브메뉴에 대한 선택을 저장하는 변수

출력:

printf함수를 통해 출력, 함수가 호출되어 대화출력

코드 세부 설명:

사용자의 선택에 따라 scanf_s함수를 통해 choice변수에 저장된다.

Do-while루프가 시작되고, 사용자가 3을 선택하기 전까지 메뉴를 계속해서 출력하고 선택지를 받는다.

선택한 항목에 따라 다른 작업을 수행하고, printWord 함수를 호출하여 해당 단어의 설명을 출력한다.

설명:

사용자가 메뉴를 통해 단어 카테고리를 선택하고, 그 안에서 세부적인 단어를 선택하여 해당 단어에 대한 설명을 볼 수 있도록 구현되어 있다.

선택지에 따라 루프가 계속해서 반복되며, 사용자가 프로그램을 종료하기를 원할 때까지 메뉴를 제공한다.

```

// 생활영어 단어 배우기 챗터
case 2: {
    int subChoice = 0;
    printf(" << 생활영어 단어 배우기 >> Wn");
    do {
        printf(
            "1) 일상 용어   2) 건물   3) 회화할 때 주의할 용어   4) "
            "종료하기WnWn");
        printf("선택: ");
        scanf_s("%d", &subChoice);

        switch (subChoice) {
            case 1: {
                printf(
                    "Wn1) 가다오다   2) 패딩   3) 우리동네   4) 버스 정류장   5) "
                    "후드티Wn");
                printf(
                    "6) 키보드   7) 미혼   8) 미팅   9) 요리사   10) 토스트Wn");
                printf("선택: ");
                scanf_s("%d", &cho);

                if (cho >= 1 && cho <= 10) {
                    printWord(&words[cho - 1]);
                } else {
                    printf("잘못된 선택입니다. 다시 선택하세요.Wn");
                }
                break;
            }
        }
    }
    case 2: {

```

```

        break;
    }
    case 2: {
        printf(
            "1) 화장실  2) 아파트  3) 원룸  4) 콘도  5) 빌라/맨션\n");
        printf("선택: ");
        scanf_s("%d", &cho);

        if (cho >= 1 && cho <= 5) {
            printWord(&words[cho + 9]); // 인덱스 10부터 건물 단어 시작
        } else {
            printf("잘못된 선택입니다. 다시 선택하세요.\n");
        }
        break;
    }
    case 3: {
        printf(
            "1) 한 시간  2) 이틀 전에  3) 10분 후에  4) 그냥  5) "
            "한국 사람들\n");
        printf("선택: ");
        scanf_s("%d", &cho);

        if (cho >= 1 && cho <= 5) {
            printWord(&words[cho + 14]); // 인덱스 15부터 회화할 때 주의할
            // 용어 시작
        } else {
            printf("잘못된 선택입니다. 다시 선택하세요.\n");
        }
        break;
    }
    case 4:
        printf("프로그램을 종료합니다.\n");
        break;
    default:
        printf("잘못된 선택입니다. 다시 선택하세요.\n");

```

```

        printf("잘못된 선택입니다. 다시 선택하세요.\n");
    }
} while (subChoice != 4);
break;
}
case 3:
    printf("프로그램을 종료합니다.\n");
    break;
default:
    printf("잘못된 선택입니다. 다시 선택하세요.\n");
}
} while (choice != 3);

```


메모리에 저장된 대화와 단어 데이터 파일 저장 및 동적으로 할당된 메모리를 해제

설명:

saveConversations() 함수:

conversation 배열에 저장된 대화 데이터를 conversations.txt라는 파일에 저장한다. 대화 데이터의 크기를 계산하여 해당 데이터를 파일에 쓴다.

saveWords() 함수:

words 배열에 저장된 단어 데이터를 words.txt라는 파일에 저장한다. 단어 데이터의 크기를 계산하여 해당 데이터를 파일에 쓴다.

Free() 함수:

사용하여 동적으로 할당된 conversations와 words 메모리를 해제한다. 프로그램이 종료될 때 해당 메모리가 운영체제에 반환되어 메모리 누수를 방지한다.

```
// 프로그램 종료 시 대화와 단어 데이터를 파일에 저장
saveConversations(conversations,
                  sizeof(conversations) / sizeof(conversations[0]));
saveWords(words, sizeof(words) / sizeof(words[0]));

// 동적으로 할당한 메모리 해제
free(conversations);
free(words);

return 0;
}
```

4. 테스트 결과

(1) 생활영어 배우기 챗터

첫 선택에서 1, 2, 3 중에 하나를 선택하여 사용자가 번호를 입력한다.

```
----쉽게 배우는 생활영어----
```

```
1) 회화 배우기      2) 단어 배우기      3) 종료하기
```

1. 회화 배우기 선택 시

```
<<생활 영어 회화 배우기>>
```

```
1) 인사하기  2) 물어보기  3) 주문하기  4) 종료하기
```

```
선택 : |
```

1 인사하기 선택 시

```
선택 : 1
```

```
---인사하기---
```

```
1) 인사말
```

```
2) 이름 묻기
```

```
3) 오랜만에 만날 때
```

```
|
```

1 인사말 선택 시

사람들이 대부분 실수할 수 있는 표현을 보여준 뒤 알맞은 표현을 보여주며 설명한다.

```
1
-----인사말-----
hey
----정정----
hey: 친해진 뒤 사용!
how are you: 전에 만난 적 있는 사람에게 사용!
```

2 이름 묻기 선택 시

```
2
---이름 묻기---
hi my name is jina
----정정----
hi i'm jina
```

3 오랜만에 만날 때의 경우

```
3
---오랜만에 만날 때---
nice to meet you
----정정----
good to see you again
```

2 물어보기 선택 시

선택: 2

---물어보기---

1) 첫 만남 무례한 질문 2) 친해지기 위해서 쓰는 표현 3) 맛이 어떤지 물어보기

1 첫 만남 무례한 질문 선택 시

-----첫 만남 무례한 질문-----

how old are you

What's your job

----정정----

정말 물어보고 싶을때->

May i ask you age?

what do you do for a living?

2 친해지기 위해 쓰는 표현

How old are you 대신 what do you do in your free time 등 이런 질문부터 시작해서 나중에 친해진 뒤 물어보기

-----친해지기 위해 쓰는 표현---

how old are you

----정정----

what do you do in your free time

3 맛이 어떤지 물어보기

```
---맛이 어떨지 물어보기---  
is it delicious?  
----정정----  
how does it taste?
```

3 주문하기 선택 시

```
선택: 3  
---주문하기----
```

- 1) 주문시 시간이 필요할 때
- 2) 결정을 했을 때
- 3)서비스인지 물어보기

1 주문 시 시간이 필요할 때 선택 시

```
선택: 3  
---주문하기----  
  
1) 주문시 시간이 필요할 때1  
---주문시 시간이 필요할 때---  
wait a minute  
----정정----  
atually, i think we need anoter time
```

2 결정을 했을 때 선택 시

```
3)서비스인지 물어보기2  
---메뉴 결정을 했을 때---  
I decided  
----정정----  
I made a decision
```

3 서비스인지 물어보기 선택 시

```
3) 서비스인지 물어보기3
---서비스인지 물어보기---
it's service?
----정정----
it's on us/ it's on the house
```

기능 2) 단어 배우기 챗터

2 단어 배우기 선택 시

```
선택: 4
1) 회화 배우기    2) 단어 배우기    3) 종료하기

2
<< 생활영어 단어 배우기 >>
1) 일상 용어    2) 건물    3) 회화할 때 주의할 용어    4) 종료하기

선택: |
```

생활영어 단어 배우기

1을 일상용어 선택 시

```
1) 일상 용어    2) 건물    3) 회화할 때 주의할 용어    4) 종료하기

선택: 1

1) 가다오다    2) 패딩    3) 우리동네    4) 버스 정류장    5) 후드티
6) 킥보드    7) 미혼    8) 미팅    9) 요리사    10) 토스트

선택: |
```

1번 가다오다 선택 시

```
---가다 오다---  
come: 화자와 청자가 가까워질 때 사용  
go: 반대로 멀어질 때 사용
```

2번 패딩을 선택 시

우리가 자주 사용하는 단어인 패딩을 보여주고 puffer jacket이라고 정정한다.

```
선택: 2  
---패딩-padding---  
puffer jacket
```

우리 동네

```
선택: 3  
---우리 동네-my neighboor---  
my neighborhood
```

버스 정류장

```
---버스 정류장-bus station---  
bus stop
```

후드 티

```
---후드티-hood-T---  
hoodie
```

킥보드

```
선택: 8  
---킥보드-kick board---  
kick scooter
```

미혼

```
선택: 7  
---미혼- solo---  
single
```

미팅

```
선택: 8  
---미팅-meeting---  
blind date
```

요리사

```
선택: 9  
---요리사-cooker---  
cook
```

토스트

```
8) 킥보드 7) 미혼  
선택: 10  
---토스트-toast---  
grilled sandwich
```


2를 선택 시 건물용어

1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기

선택: 2

1) 화장실 2) 아파트 3) 원룸 4) 콘도 5) 빌라/맨션

선택: |

화장실 선택 시

우리가 자주 사용하는 toilet대신 bathroom이 자연스러운 표현이다.

선택: 1

---화장실-toilet---

bathroom

1) 일상 용어 2) 건물

아파트 선택 시

선택: 2

---아파트-apart---

apartment/building/complex

1) 일상 용어 2) 건물

원룸

선택: 3

---원룸-one room---

studio apartment

콘도

선택: 4

---콘도-condo---

hotel

빌라/맨션

선택: 5
---빌라-villa/mansion---
apartment

3을 선택 시 회화할 때 주의할 용어

grilled sandwich
1) 일상 용어 2) 건물 3) 회화할 때 주의할 용어 4) 종료하기
선택: 3
1) 한 시간 2) 이틀 전에 3) 10분 후에 4) 그냥 5) 한국 사람들
선택: |

1 한시간

한시간 a hour 대신 an hour이라고 정정한다.

선택: 1
---한 시간 - a hour ---
an hour

이틀 전에

---이틀전에-two days before---
two days ago

10분 전에

---10분 후에-after 10 minutes---
in 10 minutes

그냥

---그냥-just---
just because

한국 사람들

```
---한국사람들-Korean people---  
Koreans
```

종료하기 선택 시

```
partment  
1) 일상 용어    2) 건물    3) 회화할 때 주의할 용어    4) 종료하기  
  
선택 : 4  
프로그램을 종료합니다.  
1) 회화 배우기    2) 단어 배우기    3) 종료하기  
:  
프로그램을 종료합니다.  
  
C:\Users\LG\source\repos\Project82\x64\Debug\Project82.exe(프로세스 2342  
개)이(가) 종료되었습니다(코드: 0개).  
이 창을 닫으려면 아무 키나 누르세요...|
```

5. 계획 대비 변경 사항

1) 어색한 상황의 대화 대신 정확한 단어와 표현만 보여주기

- 이전: 회화 배우기 챕터와, 단어배우기 챕터에서 어색한 표현 법을 쓰는 대화를 출력한다.
- 이후: 대화 대신 어색한 단어나 문장만 쓴다.
- 사유: 공간 부족으로 인한 문제점, 어색하거나 잘못된 표현을 사용한 대화를 계속해서 보여주면 똑같은 내용이 반복될 수 있기에 대화를 삭제했다.

2) 단어를 입력 받는 대신 자주 실수하는 단어표현 몇 가지 보여주기

- 이전: 고객이 단어를 입력하면 상황에 따른 비슷한 단어 사용을 추천하며 예시 문장을 보여준다.
- 이후: 대부분이 실수할 수 있는 단어 표현만 몇 가지 보여주고, 그에 따라서 정확한 표현을 보여준다.
- 사유: 공간 부족으로 인한 문제점으로 인해 변경하였다.

6. 느낀 점

너무 말이 빠를 때 이해하지 못하고 넘어갈 때가 많아서 수업하실 때 말을 천천히 하셨으면 좋겠습니다!