「Korea Polytechnic University」
프로그래밍 과제 노트

2019-01학기

교과목 담당교수	컴퓨터공학부 박정민
학과	컴퓨터공학과
학번	2017152049
이름	정하림

프로그래밍 과제 노트 목차

1. 온라인 강의 요약정리

1-1 Part1-12장. 문자열

■ 문자열과 포인터

- 문자열은 컴파일 과정에서 첫 번째 문자의 주소로 바뀌고 문자열의 종료는 \0으로 표현되는 널 문자를 통해 종료한다
- 문자열은 주소이므로 char * 형 변수로 가리킬 수 있다
- 같은 문자열 상수는 여러번 사용하더라도 같은 메모리주소를 공유하여 사용한다
- printf의 %s 변환문자는 포인터연산으로 '\0'이 나올 때 까지 문자열을 출력
- scanf 함수를 사용한 문자열 입력
  - %s를 사용하여 공백이 없는 연속문자를 입력받을 수 있다
  - 공백문자나 \n으로 인한 입력 종료 시 마지막에 \0을 넣어준다
  - 저장한 배열의 크기를 알지 못하여 한계 이상 입력시 메모리 침범이 발생할 수 있다
  - 나머지 문자는 stdin 버퍼에 담겨 다음 호출되는 함수의 입력에 사용한다
- gets 함수를 사용한 문자열 입력
  - 중간의 공백이나 탭문자를 포함하여 문자열 한줄을 입력한다
  - 버퍼의 개행문자를 가져오지만 저장할때는 널문자로 바꿔서 저장한다
  - 때문에 scanf와는 다르게 엔터키만 쳐도 입력을 끝낸다
- fgets 함수를 사용한 문자열 입력
  - 최대배열 크기 까지만 문자열을 입력한다
  - 버퍼에 있는 개행문자도 배열에 저장하고 널문자를 붙여 문자열을 완성한다
  - 널문자 직전에 붙은 개행문자가 거슬릴 때는 str[strlen(str)-1] = '\0'으로 처리한다.
- 표준 입력 함수의 버퍼 공유 문제
  - 표준 입력 함수들이 입력 버퍼를 공유하기 때문에 버퍼에 남은 개행문자로 인해 입력 오류가 발생할 수 있다
  - 이를 해결하기위해서 getchar(), scanf("%*c"), fgetc(stdin)함수를 사용하여 개행문자를 버려주거나 버퍼를 통째로 비우는 fflush(stdin) 또는 rewind(stdin)을 사용하여 문제를 해결한다

■ 문자열 연산 함수

- 문자열을 대입하는 strcpy 함수
  - strcpy 함수는 char배열에 문자열을 복사하는 대입 연산기능을 수행한다
  - 두 개의 인자를 받아 두 번째 인자를 첫 번째 인자에 복사해주고 첫 번째 인자에는 char 배열이나 그 배열명을 저장한 포인터만 사용할 수 있다.
  - 포인터를 사용하여 복사를 수행하고 복사가 끝난 곳에 '\0'(NULL)문자로 문자열을 마무리해준다
  - strncpy함수를 사용하여 세 번째 인자에 복사할 문자 수를 지정하여 복사할 수 있다

- 문자열의 길이를 계산하는 strlen 함수
  - strlen 함수는 첫 번째 인자인 배열에 저장된 문자열의 길이를 구해 반환한다
  - 널문자 이전까지 문자열을 새어 저장된 문자열의 실제 길이를 알 수 있다

- 문자열을 붙이는 strcat 함수
  - strcat 함수는 2개의 인자를 받아 두 번째 인자인 문자열을 첫 번째 인자인 문자배열에 이어붙여준다
  - 배열이 초기화되지 않으면 쓰레기값의 중간에 붙여넣을 가능성이 크므로 반드시 초기화 후 사용한다
  - strncat 함수를 사용하여 세 번째 인자에 붙여넣을 문자 수를 지정해 이어붙일 수 있다

- 문자열을 비교하는 strcmp 함수
  - strcmp 함수는 두 문자열의 사전 순서를 판단하여 그 결과를 반환한다
  - 첫 번째 인자와 두 번째 인자의 사전 순서에 따라 전자가 크면 1 후자가 크면 -1 같은 문자라면 0이 반환된다
  - 아스키코드를 통해 비교하고 대소문자 구별할 수 없기 때문에 사전 순서를 판단할 때는 반드시 대소문자를 일치시켜야 한다
  - 앞의 함수들과 같이 strncmp함수를 사용하여 세 번째 인자에 비교할 문자 수를 지정하여 비교할 수 있다

2. 심화문제 중점 문제 분석, 디버깅

2-1 심화 분석

■ [소스코드]를 참조하여 [실행화면]과 같은 계산기 프로그램을 작성하세요.

- 첫 번째 사진

```c
#include <stdio.h>
#pragma warning(disable:4996)

double calculate(double a, char op, double b);      // 사용자에게 입력받은 식을 계산해줄 함수
int printMenu();                                    // 메뉴를 프린팅 해줄 함수

int main() {
    FILE* fp;
    fp = fopen("ch12.txt", "at+");
    int num = 0, cnt = 0;       // num 메뉴 입력에 사용할 변수 cnt 파일 레코드 변수
    double input1, input2;      // 사용자에게 입력받을 두 값
    char op;                    // 연산을 위해 필요한 연산자

    while (1) {
        num = printMenu();

        switch (num) {      // switch case문으로 메뉴를 분기
            case 0:         // 0이면 종료를 위해 분기를 빠져나감
                break;
            case 1:         // 1이면 식을 입력받아 계산해준 후 다시 처음으로
                fprintf(stdout, "식을 입력하세요 : ");
                scanf("%lf %c %lf", &input1, &op, &input2);
```

- 두 번째 사진

```c
                fprintf(stdout, "계산결과 : %.2lf\n\n", calculate(input1, op, input2));
                fprintf(fp, "%03d ", cnt++);
                fprintf(fp, "계산결과 : %lf\n\n", calculate(input1, op, input2));
                continue;
        }
        break;
    }

    fclose(fp);
    system("notepad.exe ch12.txt");

    return 0;
}

int printMenu() {
    int input = -1; // 메뉴선택을 위한 입력 변수 0과 차별성을 두기위해 -1로 초기화
    fprintf(stdout, "------ MENU ------\n");
    fprintf(stdout, "0. 프로그램 종료\n");
    fprintf(stdout, "1. 계산기 실행\n");
    fprintf(stdout, "메뉴 선택 : ");
    scanf("%d", &input);        // 사용자에게 메뉴를 선택받음
```

- 세 번째 사진

```
double calculate(double a, char op, double b) {
    switch (op) {       // 연산자에 따라 연산하거나 오류메시지 출력
    case '+':
        return a + b;
    case '-':
        return a - b;
    case '*':
        return a * b;
    case '/':
        return a / b;
    default:
        fprintf(stdout, "올바른 식을 입력해주세요₩n");
    }
}
```

ch12.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

000 계산결과 : 13.333333

001 계산결과 : 3.333333

C:₩Users₩What is that₩Desktop₩HaLim₩Deeper₩Debug₩Deep

```
------ MENU ------
0. 프로그램 종료
1. 계산기 실행
메뉴 선택 : 1
식을 입력하세요 : 40 / 3
계산결과 : 13.33

------ MENU ------
0. 프로그램 종료
1. 계산기 실행
메뉴 선택 : 1
식을 입력하세요 : 10 / 3
계산결과 : 3.33

------ MENU ------
0. 프로그램 종료
1. 계산기 실행
메뉴 선택 : 0
```

지역

이름	값
cnt	2
fp	0x014e5300 {_Placeholder=0x014ef06b }
input1	10.000000000000000
input2	3.0000000000000000
num	0
op	47 '/'

자동 지역 조사식 1

-

■ 위의 문제에서 작성한 프로그램은 사용자가 숫자가 아닌 문자를 입력 했을 때 무한루프가 돌게 됩니다. 이를 해결해 [실행화면] 과 같이 실행될 수 있도록 프로그램을 수정하세요.

```
switch (num) {      // switch case문으로 메뉴를 분기
    case 0:         // 0이면 종료를 위해 분기를 빠져나감
        break;
    case 1:         // 1이면 식을 입력받아 계산해준 후 다시 처음으로
        fprintf(stdout, "식을 입력하세요 : ");
        scanf("%lf %c %lf", &input1, &op, &input2);
        fprintf(stdout, "계산결과 : %.2lf₩n₩n", calculate(input1, op, input2));
        fprintf(fp, "%03d ", cnt++);
        fprintf(fp, "계산결과 : %lf₩n₩n", calculate(input1, op, input2));
        continue;
    default:        // 메뉴에서 생기는 무한루프를 막기위해 메뉴를 담당하는 swich문에 default에 삽입
        fprintf(stdout, "올바른 메뉴번호를 입력해주세요 ₩n₩n");
        rewind(stdin);      // 키보드 입력 버퍼를 초기화 시키고 다시 입력을 받을 수 있도록 해주는 함수
        continue;
}
```

-

■ **사용자로부터 문자열을 입력받아 입력받은 문자열을 거꾸로 출력해주는 프로그램을 작성하세요. 사용자에게 최대 255자를 입력받습니다.**

```c
#include <stdio.h>
#include <string.h>
#pragma warning(disable:4996)
#define MAX 256

int main() {
    FILE* fp;
    fp = fopen("ch12.txt", "at+");
    int i = 0;      // 반복문에 사용할 i
    char input[MAX];        // 255자까지 입력받을 수 있는 배열 MAX

    fprintf(stdout, "문자열을 입력하세요 : ₩n");
    scanf("%s", input);
    fprintf(stdout, "뒤집은 문자열 : ₩n");
    // 문자열을 뒤집기위해 strlen으로 배열의 길이를 구한 후 배열의 맨뒤 널문자 직전부터 맨 처음까지 반복할 반복문
    for (i = strlen(input) - 1; i > -1; i--) {
        //반복하여 화면과 텍스트에 문자 출력
        fprintf(stdout, "%c", input[i]);
        fprintf(fp, "%c", input[i]);
    }
```

-

```c
    fprintf(fp, "₩n₩n");

    fclose(fp);
    system("notepad.exe ch12.txt");

    return 0;
}
```

ch12.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

olleh

지역

이름	값
fp	0x001849c8 {_Placeholder=0x0018c057}
i	-1
input	0x00e3fcf8 "hello"

자동 지역 조사식 1

선택 C:₩Users₩What is that₩Desktop₩HaLim₩Deep

```
문자열을 입력하세요 :
hello
뒤집은 문자열 :
olleh
```

■ **사용자로부터 문자열을 입력 받아 공백 단위로 쪼개어 재출력해주는 프로그램을 작성하세요.**

-

```c
#include <stdio.h>
#include <string.h>
#pragma warning(disable:4996)
#define MAX 256

int main() {
    FILE* fp;
    fp = fopen("ch12.txt", "at+");
    int i = 0;          // 반복문에 이용할 i 변수
    char input[MAX];    // 사용자에게 MAX-1만큼의 문자열을 입력받음

    fprintf(stdout, "문자열을 입력 : ₩n");
    gets(input);        // 공백도 입력받기 위해 gets 사용

    fprintf(stdout, "결과 : ₩n");
    for (i = 0; i < strlen(input); i++) {       // strlen 함수로 input배열의 길이만큼 반복하고 만약 공백문자가 있다면 엔터로 치환
        if (input[i] == ' ')
            input[i] = '₩n';
    }

    // 결과값을 화면과 파일에 출력해줌
    fprintf(stdout, "%s₩n", input);
    fprintf(fp, "%s₩n", input);

    fprintf(fp, "₩n₩n");
    fclose(fp);
    system("notepad.exe ch12.txt");

    return 0;
}
```

지역

이름	값
▷ fp	0x00c649c8 {_Placeholder=0x00000000 }
i	13
▷ input	0x00affa8c "What\nis\nthat?"

ch12.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

What
is
that?

C:₩Users₩What is that₩Desktop₩HaLim₩Deeper₩Debug₩Deeper.exe

```
문자열을 입력 :
What is that?
결과 :
What
is
that?
```

3-1 10주차-자기성찰 및 평가

■ 수업 및 실습을 통해서 배운 내용

- 문자열에 대한 이해
- 각종 입력 함수의 특징과 사용법
- 입력 버퍼 공유로 인한 문제점과 해결방법
- string.h에 있는 각종 문자열 연산 함수의 사용법
- 문자열 연산함수를 사용할 때 주의할 점과 원하는 개수를 지정하는 법

■ 느낀점 (자유롭게 기술)

- 문자열 상수를 char형 포인터에 넣는 것은 생각해본적 없어서 신기하고 흥미로웠다
- 저번에 느꼈던 \n 개행문자로 인해 scanf가 동작하지 않는 현상에 대해 발생 원인과 자세한 해결방법을 알 수 있어서 좋았다
- fflush(stdin)은 교수님이 알려주신 것처럼 학교pc에서는 동작하지 않았으나 내 노트북에서는 잘 동작했다. 하지만 어디에서나 동작할 수 있도록 rewind함수를 사용해야 할 것 같다
- 문자열 연산함수는 응용하여 정말 여러 분야에 사용할 수 있는 유용한 함수인 것 같다
- 이번 과제는 관련 동영상 강의가 다음주부터 들을 수 있도록 막혀있어서 책을 주로 참고하여 작성하였는데 책을 보며 작성하니 수업 때 제대로 보지 못했던 부분을 알 수 있어서 유익했다.