자료구조실습 개인 프로젝트 최종보고서

- 프로그래밍 언어 변환기 프로그램 -

2020204019 소프트웨어학부 권혜진

1. 주제

- 컴파일러를 넘어 고급언어를 같은 레벨의 고급언어로 바꿔주는 프로그래밍 언어 변환기 프로그램 개발을 주제로 함.

2. 프로그램 목적

- 소프트웨어 기술이 발전하면서 특정 언어로 작성된 소스코드를 다른 언어로 바꿔 써야 하는 경우가 존재함.
  - ➢ 새로운 기술이 매일 등장함에 따라 서로 다른 언어의 일관성을 유지하는 것이 어려워지며, 새로운 기능이 도입되는 만큼 컴파일러도 업데이트도 필요함.

- 프로그램 소스코드를 한 언어에서 다른 언어로 바꾸는 것은 대상 언어 및 목표 언어 모두에 대한 전문 지식을 필요로 하며, 많은 비용발생이 우려됨.
  - ➢ 실제 호주연방 은행은 코볼 (Cobol)에서 자바로 변환하는 데 5년동안 약 7억5천만달러(약 9000억원)를 지출함.

- 컴퓨터계열 대학생들의 교육은 Java, Python, C, C++ 등 최소 4개의 언어를 동시에 배움. 서로 다른 프로그래밍 언어를 동시에 학습하게 되어, 모든 언어를 습득을 하는데 있어 어려움이 존재함.
  - ➢ 서로 다른 프로그래밍 언어를 즉각적으로 변환해주는 프로그램을 만들어, 시각적으로 한눈에 비교하며 각 언어 습득에 대한 올바른 이해를 추진하고자 함.

3. 개발 환경

- Visual Studio cpp 언어로 진행
- 추후 리눅스에서 cpp 파일을 가져와, 실행 컴파일러 진행

4. 실제 개발 일정

	계획서 작성 및 cpp파일 입출력 구현		header, print, printf, scanf return 0, semicolon, {}, 주석 변환		각 변수(int, list, str 등) if, else if, else for문, void function 변환		최종 보고서 작성 및 테스트 진행	
~ 11/ 23 (1주차)								
~ 11/ 30 (2주차)								
~ 12 / 7 (3주차)								
~ 12/ 12 (4주차)								

5. 프로그램 기능 및 작성코드

(1) 파일 입출력

```
int main()
{
    FILE* fp = fopen("c-program.c", "r");
    FILE* wp = fopen("py-program.py", "w+");
```

파일 입출력 FILE* 을 이용하여 읽을 c파일과 변환 후 써 줄 .py 파일을 지정합니다.

(2) Header 파일 변경

```
while (!feof(fp))
{
    fgets(str, 100, fp);
    strcpy(new_str, str);

    // header transe
    rets = strtok(str, "<");
    if (strcmp("#include ", rets) == 0)
    {
        rets = strtok(NULL, ">");

        if (strcmp("string.h", rets) == 0)
        {
            fprintf(wp, "%s₩n", "import string");
            continue;
        }
        else if (strcmp("math.h", rets) == 0)
        {
            fprintf(wp, "%s₩n", "import math");
            continue;
        }
        else if (strcmp("random", rets) == 0)
        {
            fprintf(wp, "%s₩n", "import random");
            continue;
        }
        else continue;
    }
```

feof 함수로 파일 fp가 EOF를 검출했는지 확인하고, 파일의 내용을 한문 장씩 계속해서 읽습니다.

#include <string.h>헤더파일에 존재하는 strtok함수를 이용하여, 각 문자열을 기준 토큰으로 잘라줍니다.

#include <string.h>헤더파일에 존재하는 strcmp함수를 이용하여 토큰으로 자른 문자열이 파이썬 문법으로 바꿔질 문자열인지 확인합니다.

헤더파일 부분은 c언어와 파이썬 언어의 공동 헤더파일인 string, math, random 정도만 변환하였습니다.

(3) Void 함수 변경

```
if (strcmp("void", rets_two) == 0)
{
    nullspace = 1;
    fprintf(wp, "₩ndef ");
    rets_two = strtok(NULL, "(");
    fprintf(wp, "%s(",rets_two);
    rets_two = strtok(NULL, ",");
    cout <<"before count : " <<count << endl;

    while (rets_two != NULL)
    {
        rets_two = strtok(NULL, ",");

        if (rets_two == NULL)
        {
            count--;
            break;
        }
        count++;
    }

    if(count == 0)
    {
        fprintf(wp,") : ");
    }
else
{
    count++;
    rets_two = strtok(new_str3, "(");
    for (int i = 0; i < count; i++)
    {
        strtok(NULL, ",");
        if (i == count - 1)
        {
            rets_two = strtok(NULL, ")");
            fprintf(wp, "%s) : ", rets_two)
            break;
        }
        rets_two = strtok(NULL, ",");
        fprintf(wp, "%s,", rets_two);
    }
}
cout << count << endl;
count = 1;
}
```

"(" 를 기준으로 함수 명을 가져와 "def 함수명 (" 으로 우선 출력해줍니다.

strcpy함수를 이용하여, 토큰으로 자르지 않은 기존 문자열을 new_str3에 복사해주고, "(" 뒤에 , 의 개수를 세어 인자가 몇 개 들어올 것인지 확인해주며 변수 명만 가져와서 출력해줍니다.

(4) Semicolon, return 0, comment 변경

```
//semicolen transe
rets = strtok(str, ";");
string notspace(rets);
notspace.erase(notspace.begin());
vector<char> vc(notspace.begin(), notspace.end());
vc.push_back('₩0');
rets = &*vc.begin();
vc.pop_back();

// return 0 transe
if (strcmp("return 0", rets) == 0)
    continue;

rets = strtok(str, "(");
notspace = rets;
notspace.erase(notspace.begin());
vector<char> vc1(notspace.begin(), notspace.end());
vc1.push_back('₩0');
rets = &*vc1.begin();
vc1.pop_back();
```

```
rets = strtok(str, " ");
notspace = rets;
notspace.erase(notspace.begin());
vector<char> vc2(notspace.begin(), notspace.end());
vc2.push_back('₩0');
rets = &*vc2.begin();
vc2.pop_back();

//comment transe
if (strcmp("/",rets)==0)
{
    rets = strtok(NULL, "₩0");
    fprintf(wp,"₩n# %s",rets);
```

C 언어는 문장의 끝이 거의 semicolon으로 끝납니다. 하지만 파이썬은 semicolon이 필요 없습니다. 따라서, 우선 세미콜론을 기준으로 토큰을 자릅니다.

이때 strcmp함수로 문장 비교 시, 공백들을 모두 지워주고 비교해주기 위해서

Cpp 의 #include <string>에 있는 erase를 이용하여, 공백을 모두 지워줍니다.

이후 vector를 이용하여, 공백을 지워준 문자열 마지막에 '0'을 추가해줍니다.

Vector를 이용하는 이유는 strtok 토큰을 자를 시, char * 형식이 필요하기 때문입니다.

세미콜론 이후, return 0 를 만나면 continue 넘어가도록 구현하였습니다.

이후 c언어의 int main() 에서 괄호로 토큰을 쪼개어 이후 메인 함수에서의 동작들을 변환해줍니다. 우선 c언어의 주석처리 //는 #으로 바꿔주는 작업을 진행합니다. 이미 //으로 토큰을 잘라주었기 때문에 문자열의 끝인 '₩0'으로 한번 더 잘라주어, 주석 내용을 삽입해줍니다.

(5) 변수 (int double float double list str boolean 등) 변경

```
}//variable transe
else if (strcmp("int", rets) ==0 || strcmp("float", rets) == 0
    || strcmp("double", rets) == 0 || strcmp("char", rets) == 0
    || strcmp("boolean",rets) == 0 )
{
    nullspace = 0;
    rets = strtok(NULL, "=");
    notspace = rets;

    if ( notspace.find("[") != string::npos )
    {
        // list transe
        rets = strtok(rets, "[");
        fprintf(wp, "₩n%s", rets);
        if (strstr(new_str,"{") != NULL)
        {
            rets = strtok(new_str, "{");
            rets = strtok(NULL, "}");
            cout << rets << endl;
            fprintf(wp, " = [%s]", rets);
        }
        else //char[] transe
        {
            rets = strtok(new_str, "]");
            rets = strtok(NULL, ";");
            fprintf(wp, "%s", rets);
        }
    }
```

```
else {
    fprintf(wp, "₩n%s", rets);
    rets_two = strtok(new_str, "=");
    rets_two = strtok(NULL, ";");

    notspace = rets_two;
    notspace.erase(notspace.begin());
    vector<char> vc2(notspace.begin(), notspace.end());
    vc2.push_back('₩0');
    rets_two = &*vc2.begin();
    vc2.pop_back();

    // true -> True transe
    // false -> Fasle transe
    if(strcmp("true",rets_two)==0)
        fprintf(wp, "= True");
    else if (strcmp("false", rets_two) == 0)
        fprintf(wp, "= False");
    else
        fprintf(wp, "= %s", rets_two);
}
```

c언어는 변수 타입 변수명 = 값 이렇게 변수를 지정해줍니다. 반면, 파이썬은 변수명 = 값 만 필요합니다. 따라서, 토큰 = 을 기준으로 변수명과 값을 가져와 출력해줍니다.

토큰 = 을 기준으로 쪼갰을 때, list나 char [] 형태인 경우, [토큰으로 쪼개어 변수명을 가져옵니다. 이후, list는 { 를 기준으로 리스트 값을 가져와 [값] 형태로 출력하고, char [] 형태는 " 토큰을 기준으로 값을 가져와 출력해줍니다.

또한 토큰 = 기준으로 쪼갰을 때, true 또는 false 값 인 경우에는 Boolean 타입으로 인식됩니다. boolean타입은 c언어는 true 파이썬은 True로, 각 앞 문자를 대문자로 바꾸어 출력해줍니다.

(6) Print, printf 변경

```
}// print transe
else if (strcmp("print", rets) == 0)
{
    if(nullspace == 1)
    {
        fprintf(wp, "₩n₩t%s", "print(");
        rets = strtok(new_str, "(");
        rets = strtok(NULL, ")");
        fprintf(wp, "%s)", rets);
    }
    else {
        fprintf(wp, "₩n%s", "print(");
        rets = strtok(new_str, "(");
        rets = strtok(NULL, ")");
        fprintf(wp, "%s)", rets);
    }
}
```

```
//printf transe
else if (strcmp("printf", rets) == 0)
{
    if (nullspace == 1)
    {
        strcpy(new_str2, new_str);
        fprintf(wp, "₩n₩tprint(");
        rets = strtok(new_str, "(");
        rets = strtok(NULL, "%");

        rets_two = strtok(new_str2, ",");
        rets_two = strtok(NULL, ")");
        fprintf(wp, "%s{0}₩".format(%s))₩n", rets, rets_two);
    }
    else {
        strcpy(new_str2, new_str);
        fprintf(wp, "₩n%s", "print(");
        rets = strtok(new_str, "(");
        rets = strtok(NULL, "%");

        rets_two = strtok(new_str2, ",");
        rets_two = strtok(NULL, ")");
        fprintf(wp, "%s{0}₩".format(%s))₩n", rets, rets_two);
    }
```

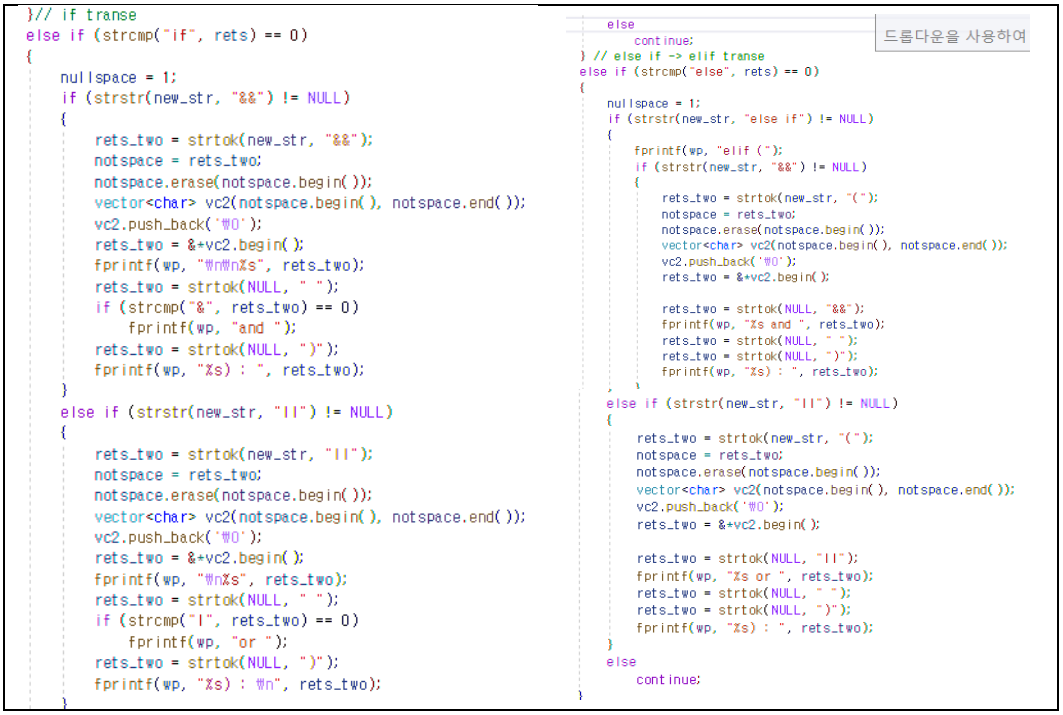
C 언어의 print문과 변수 값을 넣어줄 수 있는 printf문 2가지가 있습니다. print문의 경우에는 (괄호를 토큰으로 쪼개어 출력해주었습니다. printf문의 경우에는 %를 기준으로 파이썬의 format 형태 { }로 바꾸어 출력해줍니다. 이후 .format과 변수명까지 출력해줍니다.

(7) scanf 변경

```
} // scanf -> input transe
else if (strcmp("scanf", rets) == 0)
{
    nullspace = 0;
    rets_two = strtok(new_str,"&");
    rets_two = strtok(NULL, ")");
    fprintf(wp, "₩n%s", rets_two);
    rets_two = strtok(new_str, "%");
    rets_two = strtok(NULL, "₩"");
    if (strcmp("d", rets_two) == 0)
        fprintf(wp, " = int(input())");
    else if (strcmp("f", rets_two) == 0)
        fprintf(wp, " = float(input())");
    else
        fprintf(wp, " = input()");
```

c언어의 %d %f 인 경우에만 int(input()) float(input())으로 바꿔주는 형태로 출력하였습니다. 토큰 &으로 나누어 변수명을 가져와 먼저 출력해줍니다. String 인 경우에는 변수 타입에 관계없으므로 input() 만을 출력해줍니다.

(8) If else if else 변경



```
}// if transe
else if (strcmp("if", rets) == 0)
{
    nullspace = 1;
    if (strstr(new_str, "&&") != NULL)
    {
        rets_two = strtok(new_str, "&&");
        notspace = rets_two;
        notspace.erase(notspace.begin());
        vector<char> vc2(notspace.begin(), notspace.end());
        vc2.push_back('₩0');
        rets_two = &+vc2.begin();
        fprintf(wp, "₩n₩n%s", rets_two);
        rets_two = strtok(NULL, " ");
        if (strcmp("&", rets_two) == 0)
            fprintf(wp, "and ");
        rets_two = strtok(NULL, ")");
        fprintf(wp, "%s) : ", rets_two);
    }
    else if (strstr(new_str, "||") != NULL)
    {
        rets_two = strtok(new_str, "||");
        notspace = rets_two;
        notspace.erase(notspace.begin());
        vector<char> vc2(notspace.begin(), notspace.end());
        vc2.push_back('₩0');
        rets_two = &+vc2.begin();
        fprintf(wp, "₩n%s", rets_two);
        rets_two = strtok(NULL, " ");
        if (strcmp("|", rets_two) == 0)
            fprintf(wp, "or ");
        rets_two = strtok(NULL, ")");
        fprintf(wp, "%s) : ₩n", rets_two);
    }
    else
        continue;
} // else if -> elif transe
else if (strcmp("else", rets) == 0)
{
    nullspace = 1;
    if (strstr(new_str, "else if") != NULL)
    {
        fprintf(wp, "elif (");
        if (strstr(new_str, "&&") != NULL)
        {
            rets_two = strtok(new_str, "(");
            notspace = rets_two;
            notspace.erase(notspace.begin());
            vector<char> vc2(notspace.begin(), notspace.end());
            vc2.push_back('₩0');
            rets_two = &+vc2.begin();

            rets_two = strtok(NULL, "&&");
            fprintf(wp, "%s and ", rets_two);
            rets_two = strtok(NULL, " ");
            rets_two = strtok(NULL, ")");
            fprintf(wp, "%s) : ", rets_two);
        }
        else if (strstr(new_str, "||") != NULL)
        {
            rets_two = strtok(new_str, "(");
            notspace = rets_two;
            notspace.erase(notspace.begin());
            vector<char> vc2(notspace.begin(), notspace.end());
            vc2.push_back('₩0');
            rets_two = &+vc2.begin();

            rets_two = strtok(NULL, "||");
            fprintf(wp, "%s or ", rets_two);
            rets_two = strtok(NULL, " ");
            rets_two = strtok(NULL, ")");
            fprintf(wp, "%s) : ", rets_two);
        }
        else
            continue;
    }
```

if 문 변경에서는 우선 c 언어의 조건문에서 && 와 || 는 파이썬에서 and 와 or 로 변환되어야 합니다. 이를 선별하기 위해, #include <string.h> 에 내장되어 있는 strstr 함수를 이용하여, && 또는 || 문자가 있는지 확인해줍니다. 해당 문자가 존재한다면 토큰으로 쪼개어, string 형태의 notspace 그리고 vector를 이용하여, char * 형식으로 바꿔준 후, 공백을 모두 제거한 상태에서 and, or 로 변환을 합니다.

이후, c 언어에서 조건문의 범위를 알려주는 { } 중괄호 대신 파이썬의 : 으로 출력해줍니다.

Else if 문은 가장 초반에서 파일을 한 문장씩 읽을 때, " " 공백으로 나누는 작업을 진행하여, else와 else if 를 구분하기 어려워집니다. 따라서 이때에도 strstr 함수를 이용하여 else if 구문이 있는 경우로 나누어 and or로 바꾸어 출력해줍니다.

Else 문(코드 생략)은 else if 를 구분하는 영역 내에서 strstr로 else if 가 없는 문장일 때, 해당 변환을 진행해줍니다.

(9) For문 변경

```
} // for i in range(start,end,weight) transe
else if (strcmp("for", rets) == 0)
{
    nullspace = 1;
    // variable name find
    strtok(new_str, "(");
    strtok(NULL, " ");
    rets = strtok(NULL, " ");
    string variable = rets;

    strtok(NULL, " ");
    rets = strtok(NULL, " ");
    int start = *rets - 48;

    strtok(NULL, " ");
    rets = strtok(NULL, " ");
    string inequal_sign = rets;

    rets = strtok(NULL, ";");
    int end = std::stoi(rets);

    strtok(NULL, variable.c_str());
    rets = strtok(NULL, ")");
    string weight = rets;

fprintf(wp, "\nfor %s in range(", variable.c_str());

int inequal_sign_kind = inequal_sign_value(inequal_sign);
int weight_kind = weight_value(weight);

if (inequal_sign_kind == 1 && weight_kind == 1)
    fprintf(wp, "%d,%d,1) :", start, end);              // < & ++
else if (inequal_sign_kind == 2 && weight_kind == 1)
    fprintf(wp, "%d,%d,1) :", start, end+1);            // <= & ++
else if (inequal_sign_kind == 3 && weight_kind == 2)
    fprintf(wp, "%d,%d,-1) :", end, start);             // > & --
else if (inequal_sign_kind == 4 && weight_kind == 2)
    fprintf(wp, "%d,%d,-1) :", end, start-1);           // >= & --
else
    continue;
```

```
int inequal_sign_value(string inequal_sign)
{
    if (inequal_sign == "<")
        return 1;
    else if (inequal_sign == "<=")
        return 2;
    else if (inequal_sign == ">")
        return 3;
    else if (inequal_sign == ">=")
        return 4;
    else
        return -1;
}

int weight_value(string weight)
{
    if (weight == "++")
        return 1;
    else if (weight == "--")
        return 2;
    else
        return -1;
}
```

For 문은 c 언어와 파이썬 모두 시작, 종료, 가중치가 중요하다고 생각했습니다. 따라서 " " 공백문자와 ; 세미콜론 문자로 c 언어에서의 변수 (string variable) 시작(start) 부호(string inequal_sign) 종료(end) 가중치(weight) 총 5 개의 변수를 새로 만들었습니다.

Int 형인 start 와 end 는 char * 형식의 rets 를 *rets-48 을 하여, int 형으로 변환하거나, stoi 함수를 이용하여 형변환을 해주었습니다. 또한, for 문에서 가중치 값과 inequal_sign 에 따라 달라지는 시작 종료점의 케이스를 각각 나누기 위해 inequal_sign_value , weight_value 함수를 만들었습니다.

각 함수의 리턴 값을 기준으로 if 문들을 만들어, // 주석에 나와 있는 케이스 별로 시작 값 종료 값 가중치 값을 range for 문으로 변환해 주었습니다. 이때, string 에 저장되어 있던 변수명은 파일에 출력 시, c_str()내장함수를 이용하여, char * 형변환을 한 뒤, 출력하였습니다.

6. 프로그램 동작

```
heajin@heajin-virtual-machine:~$ gcc converter.cpp -o converter.out -lstdc++
heajin@heajin-virtual-machine:~$ ./converter.out
heajin@heajin-virtual-machine:~$ ls
MyMsg.h  __pycache__  convert.h      converter.out     fastBPE  msqclient.c    msqserver.c    py-program.py
Project  c-program.c  converter.cpp  examples.desktop  mkdir    msqclient.out  msqserver.out  공개
heajin@heajin-virtual-machine:~$
```

.c 파일

```
heajin@heajin-virtual-machine:~$ cat c-program.c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <random>

// comment

void function1(int a, int b)
{
        print("int two parametric void function\n");
}

void function2(double ab, double bc)
{
        print("double two parametric void function\n");
}

void function3(float sdf, float wer, float xcv)
{
        print("float three parametric void function\n");
}

void function4()
{
        print("not parametric void function\n");
}
```

.py 파일

```
}heajin@heajin-virtual-machine:~$ cat py-program.py
import string
import math
import random

# comment

def function1(a,b) :
        print("int two parametric void function\n")
def function2(ab,bc) :
        print("double two parametric void function\n")
def function3(sdf,wer,xcv) :
        print("float three parametric void function\n")
def function4() :
        print("not parametric void function\n")
```

헤더파일, 주석, 함수 변환

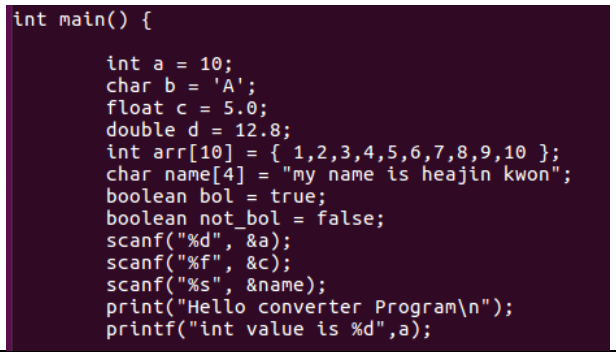
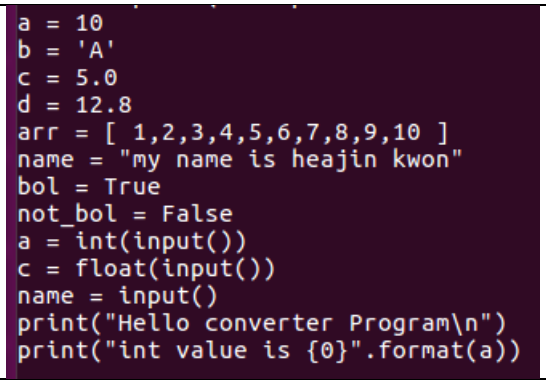
.c 파일

```
int k = 5;

if (k % 2 == 0 && k < 10)
{
printf("k is even number and one digit number k : %d",k);
}
else if (k % 2 == 0 && k >= 10)
{
printf("k is even number and not one digit number k : %d ",k);
}
else if (k % 2 == 0 || k < 10 )
{
printf("k is even number or one digit number k : %d",k);
}
else
{
printf("k : %d ",k);
}
```

.py 파일

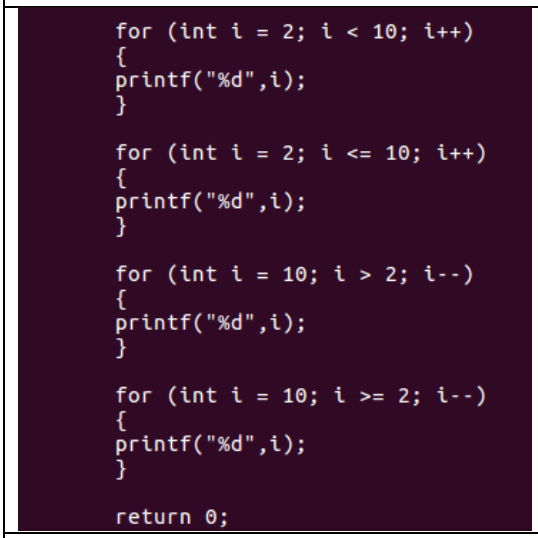
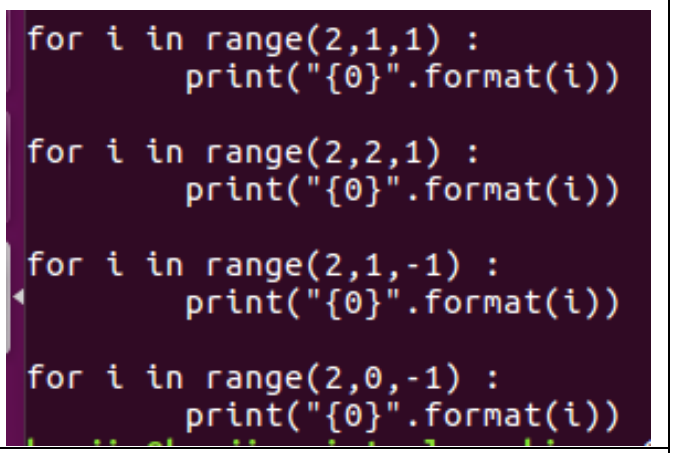
```
k = 5

if (k % 2 == 0 and k < 10) :
        print("k is even number and one digit number k : {0}".format(k))
elif (k % 2 == 0  and k >= 10) :
        print("k is even number and not one digit number k : {0}".format(k))
elif (k % 2 == 0  or k < 10 ) :
        print("k is even number or one digit number k : {0}".format(k))
else :
        print("k : {0}".format(k))
```

If else if else 문 변환

.c 파일	.py 파일
	
변수명 및 리스트 문자열 boolean print printf scanf 변환	

```c
int main() {

        int a = 10;
        char b = 'A';
        float c = 5.0;
        double d = 12.8;
        int arr[10] = { 1,2,3,4,5,6,7,8,9,10 };
        char name[4] = "my name is heajin kwon";
        boolean bol = true;
        boolean not_bol = false;
        scanf("%d", &a);
        scanf("%f", &c);
        scanf("%s", &name);
        print("Hello converter Program\n");
        printf("int value is %d",a);
```

```python
a = 10
b = 'A'
c = 5.0
d = 12.8
arr = [ 1,2,3,4,5,6,7,8,9,10 ]
name = "my name is heajin kwon"
bol = True
not_bol = False
a = int(input())
c = float(input())
name = input()
print("Hello converter Program\n")
print("int value is {0}".format(a))
```

.c 파일	.py 파일
	
For 문 변환	

```c
for (int i = 2; i < 10; i++)
{
printf("%d",i);
}

for (int i = 2; i <= 10; i++)
{
printf("%d",i);
}

for (int i = 10; i > 2; i--)
{
printf("%d",i);
}

for (int i = 10; i >= 2; i--)
{
printf("%d",i);
}

return 0;
```

```python
for i in range(2,1,1) :
        print("{0}".format(i))

for i in range(2,2,1) :
        print("{0}".format(i))

for i in range(2,1,-1) :
        print("{0}".format(i))

for i in range(2,0,-1) :
        print("{0}".format(i))
```

7. 문제 발생 및 해결방법

(1) 파일 입출력 문제

처음에는 c 파일을 읽어 txt로 저장 후, 한 줄 씩 읽으려고 했으나, 파일 입출력의 오류가 빈번하고, 이를 다시 읽어 변환 후, txt파일에 다시 쓸 때, 들여쓰기 등의 문제 발생

➔ 파이썬에서 들여쓰기가 가장 중요함. 따라서, txt 파일의 변환이 아닌 파일 자체를 읽어 변환해주고 다시 작성함.

(2) 한문 장씩 읽어 문자 비교

이때, 공백이 있는 경우, print, scanf 등 각 함수의 기능을 하는 문자열을 찾을 수 없음. 공백제거를 해주는 erase함수는 string에 존재함. erase이용 시, 다시 char* 로 변환 해야함.

➔ 기존 char * 변수를 string에 저장 후, vector를 이용하여 char * 형으로 변환

8. 과제 후기

이번 개인 프로젝트는 파일의 입출력부터 어떻게 출력을 할 것인지, 또한 한문 장씩 읽어 어떤 식으로 비교를 하며 변환을 할 것인지에 대한 해결점을 찾는 것이 가장 어려웠습니다. 처음의 이 2 가지 과정만을 해결하니, python 과 c 언어의 공통 문법, 다른 문법들을 찾아보며 해당 문법을 서로 변환하는 과정을 거칠 수 있었습니다. 처음에 변수나 헤더 파일, 주석들의 변환은 간단했으나, 아예 문법이 다른 scanf 또는 printf, for 문의 변환에서 점점 시간을 들여 고민을 하곤 했습니다. 고민하는 과정에서 c 언어와 python 의 문법차이를 꼼꼼히 공부할 수 있었으며, 여러 경우의 수를 생각하고, 이 둘의 차이점에 대해 깊이 생각해보는 시간을 가질 수 있었습니다.

무엇보다도 저는 파일을 한문 장씩 읽으며, token 으로 쪼개고, 이를 비교하는 방법에 있어서 많은 고민을 하였습니다. 이때 수업 내용을 복습하며, 배웠던 cpp 내용들을 다시 살펴보곤 했습니다. 단순히 문법을 배우는 것이 아니라 cpp 내의 컨테이너들에 대해서 배우는 수업인 자료구조실습내용에서 string 컨테이너와 vector 컨테이너를 다시 되짚어보게 되었습니다. 인터넷을 이용하여 더 구체적으로 검색해보니 string 에는 공백을 제거할 수 있는 erase 라는 함수가 이미 내장되어 있었고, 이후 vector 컨테이너를 이용하여, 기존의 char * 형으로 변환해 줄 수 있을 것이라 생각하게 되었습니다. 또한 이러한 방법이 제가 찾아본 유사 프로젝트 github 내용에서도 존재하였습니다. 이처럼 배운 내용을 기반으로 더 공부하고, 이를 적용하여 프로젝트를 진행할 수 있었습니다.

9. 참고자료

DAN-329/C_to_Python_translator: Using File I/O we were able to convert C code written in one text file to Python code in another text file with the application of multiple function that could identify and accordingly process specific key words and formats used in the C language. (github.com)

➔ C 코드를 python 코드로 변환한 예제 :

c언어 -> python 어떤 문법을 바꾼 것인지 참고함.

ojss/CtoCPP: Simple C to CPP converter (github.com)

➔ 파일 입출력부분을 참고함.