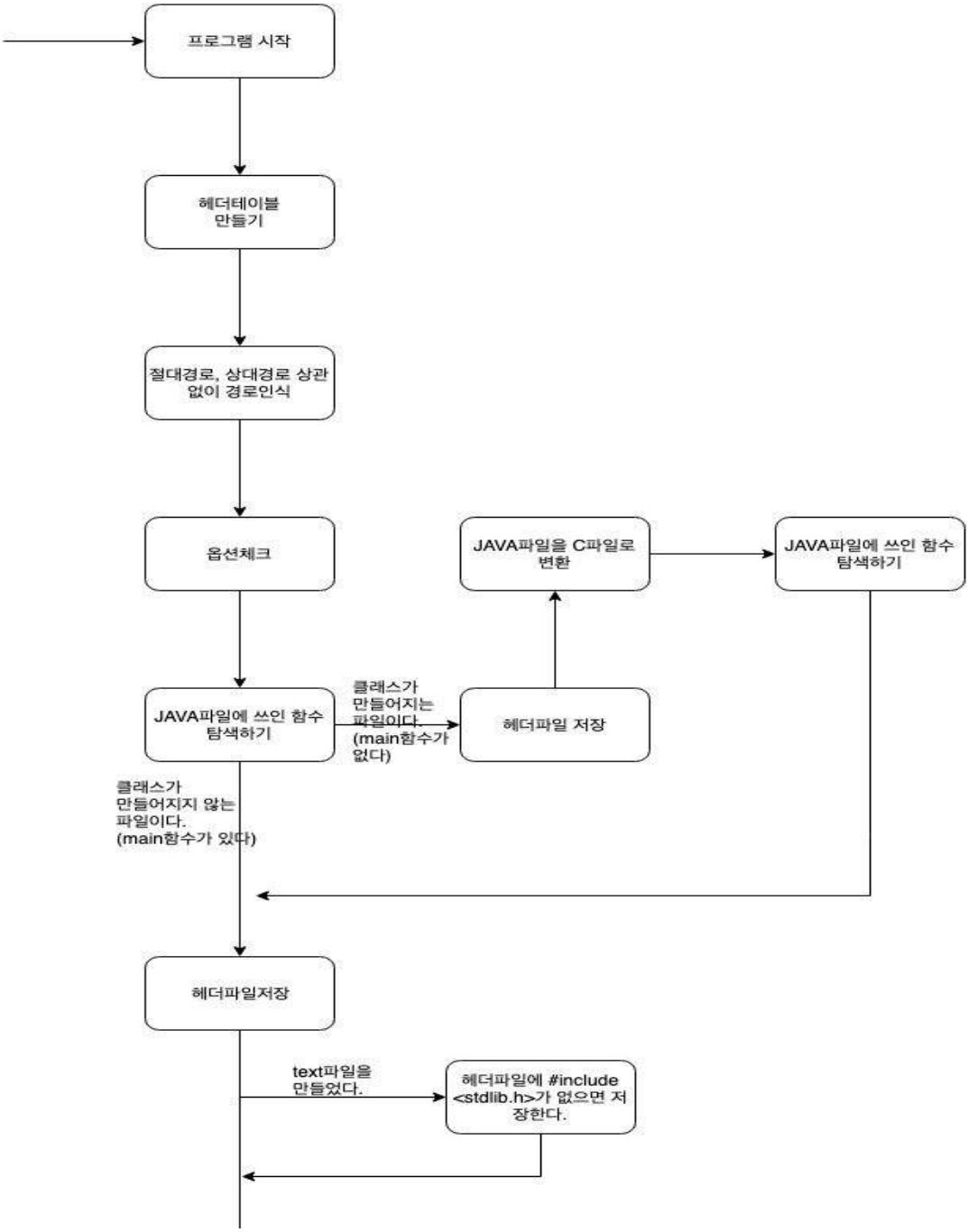
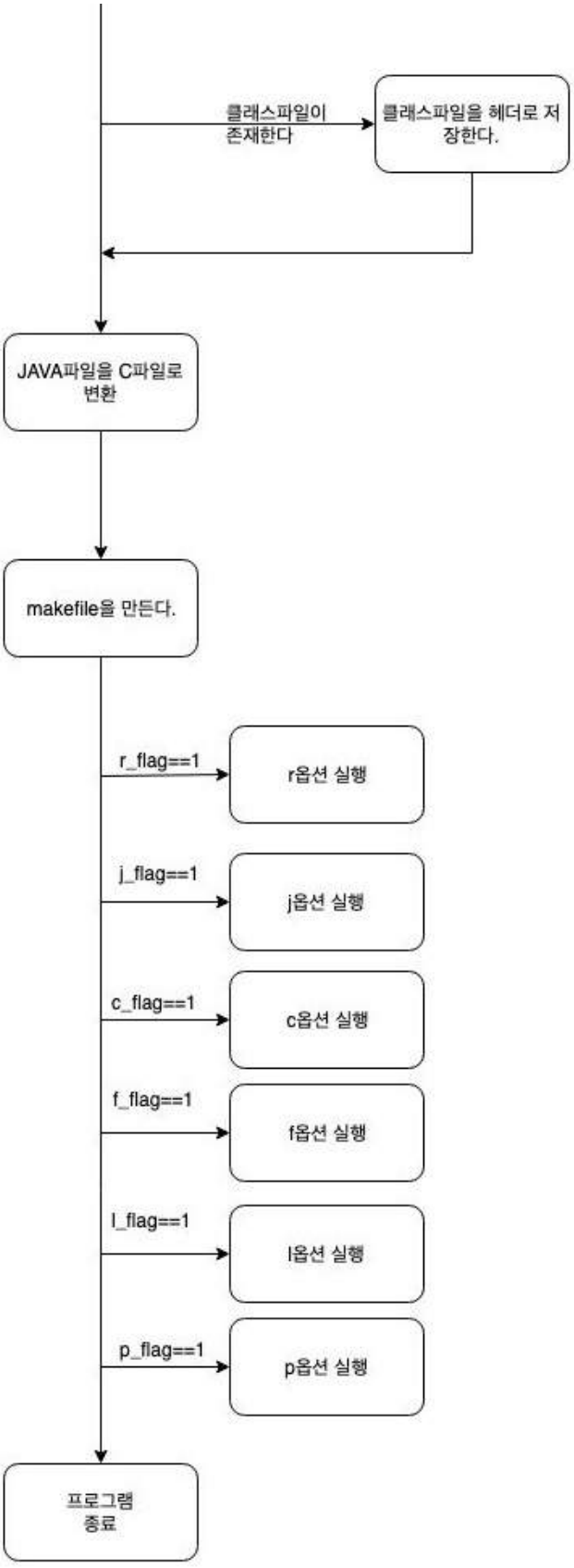
it융합학과 20160433 김민정

1. 과제 개요

ssu_convert는 JAVA언어 프로그램을 C언어 프로그램으로 자동변환해주는 프로그램이다. JAVA 파일의 내용을 한줄씩 읽으면서 그것과 같은 기능을 하게끔 C언어로 바꾸어 .c파일에 기록해준다. 헤더테이블을 먼저 만들어 C언어에 쓰인 함수에 따라 헤더파일을 C파일에 인클루드해주고 이 헤더테이블은 개발자가 수정가능하다. 여러 가지 옵션을 이용하여 자바파일을 어떻게 c파일로 변환했는지, 자바의 함수를 어떤 c언어의 함수로 바꿨는지, 자바파일과 c언어파일의 라인수, 파일크기 등을 알 수 있다.

2. 설계




클래스파일이
존재한다
클래스파일을 헤더로 저장한다.
JAVA파일을 C파일로 변환
makefile을 만든다.
r_flag==1
r옵션 실행
j_flag==1
j옵션 실행
c_flag==1
c옵션 실행
f_flag==1
f옵션 실행
l_flag==1
l옵션 실행
p_flag==1
p옵션 실행
프로그램 종료

3. 구현
void makeMakefile(char *file_name);
makefile을 만드는 함수이다.

void makeHeaderTable();
파일로 저장해놓은 헤더테이블을 읽어와 함수들과 그에 해당하는 헤더를 배열로 만드는 함수이다.

int check_option(int argc, char*argv[]);
getopt를 이용해서 옵션을 체크한다. 해당하는 옵션의 flag값을 변화시키는 함수이다.

void funcSearch(char*fname);
java파일에 어떤 함수가 쓰였는지 먼저 체크하는 함수이다. 쓰인 java함수들과 그에 대응하는 c함수들을 배열로 저장한다.

void headerfile(char*func);
함수에 따른 헤더파일을 불러와 헤더파일을 저장하는 함수

void optionP(int p_flag);
p옵션을 실행할 때 수행되는 함수이다.

void optionL(int l_flag, char*cPath, char*c_fname, char*jPath, char*j_fname);
l옵션을 실행할 때 수행되는 함수이다.

void optionJ(int j_flag, char* path,char* fname);
j옵션을 실행할 때 수행되는 함수이다.

void optionC(int c_flag, char* path,char* fname);
c옵션을 실행할 때 수행되는 함수이다.

void optionR(int r_flag, char *cPath, char* c_name, char* jPath, char*j_name);
r옵션을 실행할 때 수행되는 함수이다.

void optionF(int f_flag, char*cPath, char*c_fname, char*jPath, char*j_fname);
f옵션을 실행할 때 수행되는 함수이다.

void addTab(int box, FILE* fd);
괄호의 개수에 따라 탭을 추가해주는 함수이다.

void convertJavaToC(char* fname, char* c_file_name);
자바를 c로 변환해서 c파일로 저장하는 함수이다.

void EraseFrontSpace(char *j_str_read);
읽은 문자열들의 앞에 있는 tab을 제거해주는 함수이다.

4. 테스트 및 결과

1) make파일을 이용하여 ssu_convert를 컴파일한다.

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ make -f make
gcc -c -o ssu_convert.o ssu_convert.c
gcc -o ssu_convert ssu_convert.o
```

2) 옵션 없을 때 각각의 파일들을 컴파일하면 만들어지는 파일들이다.

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q1.java
q1.c converting is finished!
Runtime: 0:073480(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ls
h_table_file.txt  q1.java      q3.java        ssu_convert.o
make              q1_Makefile  ssu_convert    ssu_runtime.h
q1.c              q2.java      ssu_convert.c
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q2.java
q2.c converting is finished!
Runtime: 0:111662(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ls
h_table_file.txt  q1.java      q2.java      ssu_convert    ssu_runtime.h
make              q1_Makefile  q2_Makefile  ssu_convert.c  Stack.c
q1.c              q2.c         q3.java      ssu_convert.o
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q3.java
q3.c converting is finished!
Runtime: 0:077017(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ls
h_table_file.txt  q1.java      q2.java      q3.java      ssu_convert.c  Stack.c
make              q1_Makefile  q2_Makefile  q3_Makefile  ssu_convert.o
q1.c              q2.c         q3.c         ssu_convert  ssu_runtime.h
```

3) -j 옵션

-q1.java

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q1.java -j
q1.c converting is finished!
1 import java.util.Scanner;
2
3 public class q1{
4       public static void main(String[] args){
5               Scanner scn = new Scanner(System.in);
6
7               System.out.printf("Enter the number : ");
8
9               int num;
10              num = scn.nextInt();
11              int even=0, odd=0;
12
13              for(int i=1; i<=num; i++){ // Checking...
14                      if(i % 2 == 0){
15                              even+=i;
16                      }
17                      else{
```

```
18                              odd+=i;
19                      }
20              }
21
22              System.out.printf("Sum of Even number : %d\n", even);
23              System.out.printf("Sum of Odd number : %d\n", odd);
24
25              return ;
26      }
27 }
Runtime: 0:095296(sec:usec)
```

-q2.java

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q2.java -j
q2.c converting is finished!
1 class Stack{
2
3       int top;
4       int[] stack;
5
6       public static final int STACK_SIZE = 10;
7
8       public Stack(){
9               top = -1;
10              stack = new int[STACK_SIZE];
11      }
12
13      public int peek(){
14              return stack[top];
15      }
16
17      public void push(int value){
18              stack[++top] = value;
19              System.out.printf("%d PUSH !\n", stack[top]);
20      }
21
22      public int pop(){
23              System.out.printf("%d POP !\n", stack[top]);
24              return stack[top--];
25      }
26
27      public void printStack(){
28              System.out.printf("\n-----STACK LIST-----\n");
29
30              for(int i=top; i>=0; i--){
31                      System.out.printf("%d\n",stack[i]);
32              }
33
34              System.out.printf("-----END OF LIST-----\n");
35      }
36 }
37
```

```
38 public class q2{
39
40      public static void main(String args[]){
41
42              Stack st = new Stack();
43
44              st.push(5);
45              st.push(2);
46              st.push(3);
47              st.push(4);
48              st.push(1);
49
50              st.printStack();
51
52              st.pop();
53              st.pop();
54              st.push(15);
55
56              System.out.printf("TOP OF STACK : %d\n", st.peek());
57
58              st.printStack();
59
60              st.pop();
61              st.pop();
62              st.pop();
63              st.pop();
64
65              st.push(30);
66
67              st.printStack();
68      }
69 }
Runtime: 0:126990(sec:usec)
```

-q3.java

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q3.java -j
q3.c converting is finished!
1 import java.io.File;
2 import java.io.IOException;
3 import java.io.FileWriter;
4
5 public class q3{
6       public static void main(String[] args) throws IOException{
7
8
9               File file = new File("q3java.txt");
10
11              /****** 두번째 매개변수 ******/
12              /****** true : 기존 파일의 내용 이후부터 쓰여짐 ******/
13              /****** false : 처음부터 쓰여짐 ******/
14
15              FileWriter writer = new FileWriter(file, false);
16
17              writer.write("2019 OSLAB\n");
18              writer.write("Linux System Programming\n");
19
20              writer.flush();
21              System.out.printf("DONE\n");
22
23              if(writer != null)
24                      writer.close();
25
26      }
27 }
Runtime: 0:104740(sec:usec)
```

4) -c 옵션

-q1.java

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q1.java -c
q1.c converting is finished!
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5
6       printf("Enter the number : ");
7
8       int num;
9       scanf("%d",&num);
10      int even=0, odd=0;
11
12      for(int i=1; i<=num; i++){ // Checking...
13              if(i % 2 == 0){
14                      even+=i;
15              }
16              else{
17                      odd+=i;
18              }
19      }
20
21      printf("Sum of Even number : %d\n", even);
22      printf("Sum of Odd number : %d\n", odd);
23
24      exit(0);
25 }
Runtime: 0:087135(sec:usec)
```

-q2.java

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q2.java -c
q2.c converting is finished!
Stack.c
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int top;
5 int *stack = NULL;
6
7 #define STACK_SIZE 10
8
9 void Stack(){
10      top = -1;
11      stack =(int*)malloc(sizeof(int)*STACK_SIZE);
12 }
13
14 int peek(){
15      return stack[top];
16 }
17
18 void push(int value){
19      stack[++top] = value;
20      printf("%d PUSH !\n", stack[top]);
21 }
22
23 int pop(){
24      printf("%d POP !\n", stack[top]);
25      return stack[top--];
26 }
27
28 void printStack(){
29      printf("\n-----STACK LIST-----\n");
30
31      for(int i=top; i>=0; i--){
32              printf("%d\n",stack[i]);
33      }
34
35      printf("-----END OF LIST-----\n");
36 }

q2.c
1 #include <stdio.h>
2 #include "Stack.c"
3
4 int main(void){
5
6       Stack();
7
8       push(5);
9       push(2);
10      push(3);
```

```
11      push(4);
12      push(1);
13
14      printStack();
15
16      pop();
17      pop();
18      push(15);
19
20      printf("TOP OF STACK : %d\n", peek());
21
22      printStack();
23
24      pop();
25      pop();
26      pop();
27      pop();
28
29      push(30);
30
31      printStack();
32 }
Runtime: 0:145612(sec:usec)
```

-q3.java

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q3.java -c
q3.c converting is finished!
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5
6       char* file = "q3java.txt";
7
8       /****** 두번째 매개변수 ******/
9       /****** "a" : 기존 파일의 내용 이후부터 쓰여짐 ******/
10      /****** "w" : 처음부터 쓰여짐 ******/
11
12      FILE* writer = fopen(file, "w");
13      if(writer == NULL){
14              fprintf(stderr, "open error for %s\n", file);
15              exit(1);
16      }
17
18      fputs("2019 OSLAB\n",writer);
19      fputs("Linux System Programming\n",writer);
20
21      fflush(writer);
22      printf("DONE\n");
23
24      if(writer != NULL)
25              fclose(writer);
26 }
Runtime: 0:108975(sec:usec)
```

5) -p 옵션

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q1.java -p
q1.c converting is finished!
1 System.out.printf() -> printf()
2 scn.nextInt() -> scanf()
Runtime: 0:092466(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q2.java -p
q2.c converting is finished!
1 System.out.printf() -> printf()
Runtime: 0:123702(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q3.java -p
q3.c converting is finished!
1 FileWriter () -> fopen()
2 writer.write() -> fputs()
3 writer.flush() -> fflush()
4 System.out.printf() -> printf()
5 writer.close() -> fclose()
Runtime: 0:098440(sec:usec)
```

6) -f 옵션

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q1.java -f
q1.c converting is finished!
q1.java file size is 466 bytes
q1.c file size is 343 bytes
Runtime: 0:091245(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q2.java -f
q2.c converting is finished!
q2.java file size is 979 bytes
Stack.c file size is 510 bytes
q2.c file size is 281 bytes
Runtime: 0:122431(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q3.java -f
q3.c converting is finished!
q3.java file size is 589 bytes
q3.c file size is 515 bytes
Runtime: 0:114995(sec:usec)
```

7) -l 옵션

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q1.java -l
q1.c converting is finished!
q1.java line number is 27 lines
q1.c line number is 25 lines
Runtime: 0:105799(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q2.java -l
q2.c converting is finished!
q2.java line number is 69 lines
Stack.c line number is 36 lines
q2.c line number is 32 lines
Runtime: 0:141504(sec:usec)
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q3.java -l
q3.c converting is finished!
q3.java line number is 27 lines
q3.c line number is 26 lines
Runtime: 0:093967(sec:usec)
```

8) -r 옵션

-q1.java

```
--------
q1.java
--------
1 import java.util.Scanner;
2
3 public class q1{
4       public static void main(String[] args){
5               Scanner scn = new Scanner(System.in);
6
7               System.out.printf("Enter the number : ");
8
9               int num;
--------
q1.c
--------
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5
6       printf("Enter the number : ");
7
8       int num;
^C
```

-q3.java

```
--------
q3.java
--------
1 import java.io.File;
2 import java.io.IOException;
3 import java.io.FileWriter;
4
5 public class q3{
6       public static void main(String[] args) throws IOException{
7
8
9               File file = new File("q3java.txt");
10
11              /****** 두번째 매개변수 ******/
12              /****** true : 기존 파일의 내용 이후부터 쓰여짐 ******/
13              /****** false : 처음부터 쓰여짐 ******/
--------
q3.c
--------
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5
6       char* file = "q3java.txt";
7
8       /****** 두번째 매개변수 ******/
9       /****** "a" : 기존 파일의 내용 이후부터 쓰여짐 ******/
10      /****** "w" : 처음부터 쓰여짐 ******/
```

-q2.java

```
--------
q2.java
--------
1 class Stack{
2
3       int top;
4       int[] stack;
5
6       public static final int STACK_SIZE = 10;
7
8       public Stack(){
9               top = -1;
10              stack = new int[STACK_SIZE];
11      }
12
13      public int peek(){
14              return stack[top];
15      }
16
17      public void push(int value){
18              stack[++top] = value;
19              System.out.printf("%d PUSH !\n", stack[top]);
20      }
21
22      public int pop(){
23              System.out.printf("%d POP !\n", stack[top]);
24              return stack[top--];
25      }
26
27      public void printStack(){
28              System.out.printf("\n-----STACK LIST-----\n");
29
30              for(int i=top; i>=0; i--){
31                      System.out.printf("%d\n",stack[i]);
32              }
33
34              System.out.printf("-----END OF LIST-----\n");
35      }
36 }
37
38 public class q2{
39
40      public static void main(String args[]){
41
42              Stack st = new Stack();
43
44              st.push(5);
45              st.push(2);
46              st.push(3);
47              st.push(4);
--------
Stack.c
```

```
47              st.push(4);
--------
Stack.c
--------
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int top;
5 int *stack = NULL;
6
7 #define STACK_SIZE 10
8
9 void Stack(){
10      top = -1;
11      stack =(int*)malloc(sizeof(int)*STACK_SIZE);
12 }
13
14 int peek(){
15      return stack[top];
16 }
17
18 void push(int value){
19      stack[++top] = value;
20      printf("%d PUSH !\n", stack[top]);
21 }
22
23 int pop(){
24      printf("%d POP !\n", stack[top]);
25      return stack[top--];
26 }
27
28 void printStack(){
29      printf("\n-----STACK LIST-----\n");
30
31      for(int i=top; i>=0; i--){
32              printf("%d\n",stack[i]);
33      }
34
35      printf("-----END OF LIST-----\n");
36 }
37 }
--------
q2.c
--------
1 #include <stdio.h>
2 #include "Stack.c"
3
4 int main(void){
5
6       Stack();
7
```

```
37 }
--------
q2.c
--------
1 #include <stdio.h>
2 #include "Stack.c"
3
4 int main(void){
5
6       Stack();
7
8       push(5);
9       push(2);
10      push(3);
11      push(4);
^C
```

9) 옵션 동시에 사용

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./ssu_convert q1.java -f -l
q1.c converting is finished!
q1.java file size is 466 bytes
q1.c file size is 343 bytes
q1.java line number is 27 lines
q1.c line number is 25 lines
Runtime: 0:090102(sec:usec)
```

10) make파일 실행한 후의 결과

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ make -f q1_Makefile
gcc -o q1 q1.c
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ make -f q2_Makefile
gcc -o q2 q2.c
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ make -f q3_Makefile
gcc -o q3 q3.c
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ls
h_table_file.txt  q1.c         q2       q2_Makefile  q3.java      ssu_convert.c  Stack.c
make              q1.java      q2.c     q3           q3_Makefile  ssu_convert.o
q1                q1_Makefile  q2.java  q3.c         ssu_convert  ssu_runtime.h
```

-q1.c 실행결과

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./q1
Enter the number : 10
Sum of Even number : 30
Sum of Odd number : 25
```

-q2.c 실행결과

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./q2
5 PUSH !
2 PUSH !
3 PUSH !
4 PUSH !
1 PUSH !

-----STACK LIST-----
1
4
3
2
5
-----END OF LIST-----
1 POP !
4 POP !
15 PUSH !
TOP OF STACK : 15

-----STACK LIST-----
15
3
2
5
-----END OF LIST-----
15 POP !
3 POP !
2 POP !
5 POP !
30 PUSH !

-----STACK LIST-----
30
-----END OF LIST-----
```

-q3.c 실행결과

```
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ./q3
DONE
minjeong@minjeong-VirtualBox:~/Desktop/share/ssu_convert$ ls
h_table_file.txt  q1.java      q2.java      q3.java      ssu_convert.c
make              q1_Makefile  q2_Makefile  q3java.txt   ssu_convert.o
q1                q2           q3           q3_Makefile  ssu_runtime.h
q1.c              q2.c         q3.c         ssu_convert  Stack.c
```

5. 소스코드

<ssu_runtime.h>

```
#include <sys/time.h>

#define SECOND_TO_MICRO 1000000

void ssu_runtime(struct timeval* begin_t, struct timeval* end_t){
        end_t -> tv_sec -= begin_t -> tv_sec;

        if(end_t -> tv_usec < begin_t -> tv_usec){
                end_t -> tv_sec--;
                end_t -> tv_usec += SECOND_TO_MICRO;
        }

        end_t -> tv_usec -= begin_t -> tv_usec;
        printf("Runtime: %ld:%06ld(sec:usec)\n", end_t -> tv_sec, end_t -> tv_usec);
}
```

<ssu_convert.c>

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/stat.h>
#include <ctype.h> //isspace()를 위해 추가한 헤더파일
#include <sys/wait.h>
#include "ssu_runtime.h“//프로그램의 실행시간을 출력해주는 함수를 포함하는 헤더파일

#define MAX_SIZE 1024
#define CONDITIONAL 4
#define JFUNC 13
#define HEADERTABLE 8

void makeMakefile(char *file_name); //makefile을 만드는 함수
void makeHeaderTable();//헤더테이블을 만드는 함수
int check_option(int argc, char*argv[]);//옵션체크해서 flag값 변화시키는 함수
void funcSearch(char*fname);//java파일에 어떤 함수가 쓰였는지 먼저 체크하는 함수
void headerfile(char*func);//함수에 따른 헤더파일을 불러와 헤더파일을 저장하는 함수
void optionP(int p_flag);//p옵션 실행시 수행되는 함수
void optionL(int l_flag, char*cPath, char*c_fname, char*jPath, char*j_fname);//l옵션 실행시 수행되는 함수
void optionJ(int j_flag, char* path,char* fname);//j옵션 실행시 수행되는 함수
void optionC(int c_flag, char* path,char* fname);//c옵션 실행시 수행되는 함수
void optionR(int r_flag, char *cPath, char* c_name, char* jPath, char*j_name);//r옵션 실행시 수행되는 함수
void optionF(int f_flag, char*cPath, char*c_fname, char*jPath, char*j_fname);//f옵션 실행시 수행되는 함
```

```
void addTab(int box, FILE* fd);//탭추가해주는 함수
void convertJavaToC(char* fname, char* c_file_name);//자바파일을 c파일로 변환해서 파일로 저장하는 함수
void EraseFrontSpace(char *j_str_read);//문자열 제일 앞에 tab 제거해주는 함수

char headerTable_func[HEADERTABLE][MAX_SIZE];//헤더테이블에서 함수를 저장하는 배열
char headerTable_header[HEADERTABLE][MAX_SIZE];//헤더테이블에서 헤더를 저장하는 배열
int headerTable_count=0;//헤더테이블에 있는 데이터의 개수
char java_func[JFUNC][MAX_SIZE]={"System.out.printf","Scanner ","return","public","FileWriter ","File ","int ","int[]
",".write",".flush",".close"};//자바의 함수 저장
char java_conditional[CONDITIONAL][MAX_SIZE]={"for","if","else","else if"};//자바에서 조건문 저장
int fileType=-1; //main함수 안의 내용인지 아닌지를 판단하는 변수
//==-1 =>아직 한번도 탐색안함
//==1 =>main만 있는 파일이다.
//==2 =>class가 있는 파일이다.
//==3 =>class가 있는 파일이다. 근데 아직 class파일은 안만들어졌다.
//==4 =>class가 있는 파일인데 이제 main함수를 볼거다.
//==5 =>class가 있는 파일의 main함수 보는중
char txt_filevar[MAX_SIZE];//텍스트파일 변수 이름
char txt_filename[MAX_SIZE];//텍스트파일 이름
char txt_fileDis[MAX_SIZE];//파일 디스크립터이름
int txt_file_opt;//파일이 어떤 버전으로 쓰여지는지 확인하는 변수
int txt_check=0;//텍스트 파일 만드는지 체크
char c_func[20][MAX_SIZE];//c에 쓸 함수 저장
int c_count=0;//c에 쓸 함수 개수
char c_func_class[20][MAX_SIZE];//클래스에 쓰인 c에 쓴 함수 저장
int c_count_class=0;
char printf_context[20][MAX_SIZE];//printf의 내용 저장
int printf_count=0;//printf함수 개수
char printf_context_class[20][MAX_SIZE];//클래스에 쓰인 printf내용 저장
int printf_count_class=0;
char headerName[100][MAX_SIZE];//내가 사용할 헤더파일 저장하는 배열
int count=0;//헤더파일 개수 저장하는 변수
//int count_class=0;
char use_java_func[20][MAX_SIZE];//java에서 실제로 쓴 함수들
char use_c_func[20][MAX_SIZE];//c에서 실제로 쓸 함수들
int use=0;
char use_java_func_class[20][MAX_SIZE];//클래스에 쓰인 java에서 실제로 쓴 함수들
char use_c_func_class[20][MAX_SIZE];//클래스에 쓰인 c에서 실제로 쓸 함수들
int use_class=0;
char var[10][MAX_SIZE];//java에 있는 변수 저장
int var_count=0;
char var_class[10][MAX_SIZE];//클래스에 쓰인 java에 있는 변수 저장
int var_count_class=0;
char var_init[10][MAX_SIZE];//java에 있는 초기화된 변수 저장
```

```
int var_init_count=0;
char var_init_class[10][MAX_SIZE];//클래스에 쓰인 java에 있는 초기화된 변수 저장
int var_init_count_class=0;
char class_name[MAX_SIZE];//class이름 저장
char use_constructor[MAX_SIZE];//생성자저장
int constructor_check=0;
char use_myfunc[10][MAX_SIZE];//사용자생성함수 저장
int myfunc_count=0;
char use_macro[10][MAX_SIZE];//매크로 저장
int macro_count=0;
char macro_define[10][MAX_SIZE];//매크로 값 저장
char use_arr[10][MAX_SIZE];//배열이름 저장
int arr_count=0;
int exp_class=0;//class파일 만들었는지 체크.
int cl_check=0;//main시작할 때 생성자 넣었는지 체크
int arr_check=0;
int b_check=0;//{가 바로 옆에 붙어있는지 확인
int bc_check=0;//}가 바로 옆에 붙어있는지 확인
int no_box_check=0;//{}가 없을 때 체크
int no_box_check2=0;
int j_flag=0;//j옵션 실행시 1
int c_flag=0;//c옵션 실행시 1
int p_flag=0;//p옵션 실행시 1
int f_flag=0;//f옵션 실행시 1
int l_flag=0;//l옵션 실행시 1
int r_flag=0;//r옵션 실행시 1
int jLine=0;//java파일 라인저장하는 변수
int cLine=0;//c파일 라인저장하는 변수

int main(int argc, char* argv[]){
        struct timeval begin_t, end_t;
        gettimeofday(&begin_t, NULL);
        makeHeaderTable();//헤더테이블 만들기
        char fname[MAX_SIZE];
        char j_name[MAX_SIZE];

        //경로에서 파일이름만을 빼내서 저장하기
        strcpy(j_name,argv[1]);
        char dummy[MAX_SIZE];
        strcpy(dummy,j_name);
        char *filename[1];
        strtok_r(dummy,".",&filename[0]);
        strcpy(filename[0],dummy);
```

```
//절대경로, 상대경로 상관없이 경로인식
if(argv[1][0]=='/')
	sprintf(fname,"%s",argv[1]);
else
	sprintf(fname,"./%s",argv[1]);

//option check
if(!check_option(argc,argv))
	exit(1);

char c_file_name[MAX_SIZE]; //c파일로 저장할 이름 저장

//Java에 있는 함수들 탐색해서 c언어의 어떤 함수로 바꿀지 결정한다.
funcSearch(fname);

//클래스가 있는 파일이다
if (fileType == 3) {
	//main없는 파일 만들기
	//헤더파일 저장
	for (int i = 0; i < use; i++) {
		headerfile(use_c_func[i]);
	}
	sprintf(c_file_name, "%s.c", class_name);
	//변환하기
	convertJavaToC(fname, c_file_name);
	exp_class = 1;
	//탐색하기
	funcSearch(fname);
}

//main있는 파일 만들기
//헤더파일 저장
for(int i=0;i<use;i++){
	headerfile(use_c_func[i]);
}
int hCheck=0;

//text파일 만들었다면 예외처리를 시켜주면서 exit(1)을 썼기 때문에 #include <stdlib.h> 헤더를 추가한다.
if(txt_check){
	for(int i=0;i<count;i++){
		if(!strcmp(headerName[i],"#include <stdlib.h>")){
			hCheck=1;
		}
	}
```

```
                if(!hCheck){
                        strcpy(headerName[count],"#include <stdlib.h>");
                        count++;
                }
        }

        //클래스파일이 존재한다.
        if (exp_class) {
                //클래스 파일 헤더로 저장하기
                char classheader[MAX_SIZE];
                sprintf(classheader,"#include \"%s\"",c_file_name);
                strcpy(headerName[count],classheader);
                count++;
        }
        sprintf(c_file_name,"%s.c",filename[0]);

        //Java에 있는 함수들 탐색해서 c언어의 어떤 함수로 바꿀지 결정한다.
        convertJavaToC(fname,c_file_name);
        printf("%s converting is finished!\n",c_file_name);
        makeMakefile(filename[0]);

        //옵션함수 실행
        optionR(r_flag, c_file_name, c_file_name, fname, j_name);
        optionJ(j_flag,fname,j_name);
        optionC(c_flag,c_file_name,c_file_name);
        optionF(f_flag,c_file_name,c_file_name,fname,j_name);
        optionL(l_flag,c_file_name,c_file_name,fname,j_name);
        optionP(p_flag);

        gettimeofday(&end_t, NULL);
        ssu_runtime(&begin_t, &end_t);
        exit(0);
}

void makeMakefile(char *file_name){
        FILE* fd;
        char makefile[MAX_SIZE];
        char executeName[MAX_SIZE];
        char data[MAX_SIZE];
        sprintf(makefile,"%s_Makefile",file_name);
        sprintf(executeName,"%s",file_name);
        sprintf(data,"main:\n\tgcc -o %s %s.c",executeName, file_name);
        if((fd=fopen(makefile,"w"))==NULL){
                fprintf(stderr, "fopen error %s\n", makefile);
```

```
                exit(1);
        }
        fputs(data,fd);
        fclose(fd);
}

void makeHeaderTable(){
        //이제 여기에 헤더테이블을 만들어서 저장한다
        char path[MAX_SIZE];
        getcwd(path,MAX_SIZE);
        sprintf(path,"%s/h_table_file.txt",path);
        char context[MAX_SIZE];
        FILE* fd;
        if((fd=fopen(path,"r"))==NULL){
                fprintf(stderr, "fopen error for %s\n",path);
                exit(1);
        }
        while(1){
                fgets(context,sizeof(context),fd);
                if(feof(fd))
                        break;
                char *cut_h[1];
                char *cut_dummy[1];
                strtok_r(context," ",&cut_h[0]);
                strtok_r(cut_h[0],"\n",&cut_dummy[0]);
                strcpy(headerTable_func[headerTable_count],context);
                strcpy(headerTable_header[headerTable_count],cut_h[0]);
                headerTable_count++;
        }
        fclose(fd);
}

//괄호의 개수에 따라 탭을 추가해주는 함수이다.
void addTab(int box,FILE*fd){
        for(int i=1;i<box;i++){
                fputs("\t",fd);
        }
}

//getopt를 이용해서 옵션을 체크한다. 해당하는 옵션의 flag값을 변화시키는 함수이다.
int check_option(int argc, char*argv[]){
        int opt;
        while((opt=getopt(argc,argv,"jcpflr"))!=-1){
                switch(opt){
```

```
                        case 'j':
                                j_flag=1;
                                break;
                        case 'c':
                                c_flag=1;
                                break;
                        case 'p':
                                p_flag=1;
                                break;
                        case 'f':
                                f_flag=1;
                                break;
                        case 'l':
                                l_flag=1;
                                break;
                        case 'r':
                                r_flag=1;
                                break;
                        case '?':
                                printf("Unknown option : %c\n",optopt);
                                return 0;
                }
        }
        return 1;
}

//java파일에 어떤 함수가 쓰였는지 먼저 체크하는 함수이다.
//쓰인 java함수들과 그에 대응하는 c함수들을 배열로 저장한다.
void funcSearch(char*fname){
        FILE* fd;
        char str_read[MAX_SIZE];
        int scanf_check=0;
        int save_p=0;
        int save_s=0;
        int save_r=0;
        int save_fw=0;
        int save_dw=0;
        int save_ff=0;
        int save_fc=0;
        int first_init = 0;
        if((fd=fopen(fname,"r"))==NULL){
                fprintf(stderr, "fopen error for %s\n",fname);
                exit(1);
        }
```

```
while(1){
	fgets(str_read,sizeof(str_read),fd);
	if(feof(fd))
		break;
	//아직 한번도 public class나 class가 발견된 적 없다
	if (fileType == -1) {
		//드디어 public class 발견! -> class 없고 main만 있는 파일이다.
		if (strstr(str_read, "public class ") != NULL) {
			fileType = 1;
			continue;
		}
		//class 발견 -> class가 있는 파일이다
		else if (strstr(str_read, "class ") != NULL) {
			fileType = 2;
			if(strstr(str_read,"{")!=NULL){
				//class이름을 저장한다.
				char* cut_str_class[1];
				char* cut_str_class2[1];
				strtok_r(str_read," ",&cut_str_class[0]);
				strtok_r(cut_str_class[0],"{",&cut_str_class2[0]);
				strcpy(class_name,cut_str_class[0]);
				continue;
			}
			else{
				//class이름을 저장한다.
				char* cut_str_class3[1];
				char* cut_str_class4[1];
				strtok_r(str_read," ",&cut_str_class3[0]);
				strtok_r(cut_str_class3[0],"\n",&cut_str_class4[0]);
				strcpy(class_name,cut_str_class3[0]);
				continue;
			}
		}
	}
	//main함수만 존재하는 파일의 main내용을 탐색한다.
	else if (fileType == 1) {
		//탐색하기
		for(int i=0;i<JFUNC;i++){
			if(strstr(str_read,java_func[i])!=NULL){
				switch(i){
					//java의 printf함수 발견
					case 0 :
						strcpy(c_func[c_count],"printf");
						char *cut_str4[1];
```

```
                char *cut_str5[1];
                strtok_r(str_read,"(",&cut_str4[0]);
                strtok_r(cut_str4[0],")",&cut_str5[0]);
                strcpy(printf_context[printf_count],cut_str4[0]);
                c_count++;
                printf_count++;
                //쓰인다고 따로 배열에 저장
                if(!save_p){
                        sprintf(use_java_func[use],"%s",java_func[i]);
                        sprintf(use_c_func[use],"printf");
                        use++;
                        save_p=1;
                }
                break;
                //java의 scanf함수 발견
        case 1 :
                //약간의 가공과정
                if(scanf_check==0){
                        char *cut_str[1];
                        char *cut_str6[1];
                        strtok_r(str_read," =",&cut_str[0]);
                        strtok_r(cut_str[0], " =",&cut_str6[0]);
                        strcpy(java_func[i],cut_str[0]);
                        scanf_check=1;
                }
                else if(scanf_check==1){
                        char *cut_str2[1];
                        char *cut_str3[1];
                        strtok_r(str_read," ",&cut_str2[0]);
                        strtok_r(cut_str2[0]," ",&cut_str3[0]);
                        strtok_r(cut_str3[0],"(",&cut_str2[0]);
                        strcpy(java_func[i],cut_str3[0]);
                        strcpy(c_func[c_count],"scanf");
                        c_count++;
                        //쓰인다고 따로 배열에 저장
                        if(!save_s){

sprintf(use_java_func[use],"%s",java_func[i]);
                                sprintf(use_c_func[use],"scanf");
                                use++;
                                save_s=1;
                        }
                }
                break;
```

```
		//java에서 return발견
	case 2:
		strcpy(c_func[c_count],"exit");
		c_count++;
		//쓰인다고 따로 배열에 저장
		if(!save_r){
			sprintf(use_java_func[use],"%s",java_func[i]);
			sprintf(use_c_func[use],"exit");
			use++;
			save_r=1;
		}
		break;
		//java에서 FileWriter 발견
	case 4:
		strcpy(c_func[c_count],"fopen");
		c_count++;
		//쓰인다고 따로 배열에 저장
		if(!save_fw){
			sprintf(use_java_func[use],"%s",java_func[i]);
			sprintf(use_c_func[use],"fopen");
			use++;
			save_fw=1;
		}
		char* cut_writefile[1];
		char* cut_writefile2[1];
		strtok_r(str_read," ",&cut_writefile[0]);
		strtok_r(cut_writefile[0]," ",&cut_writefile2[0]);
		strcpy(txt_fileDis,cut_writefile[0]);
		strtok_r(cut_writefile2[0],",",&cut_writefile[0]);
		strtok_r(cut_writefile[0],")",&cut_writefile2[0]);
		if(strstr(cut_writefile[0],"false")!=NULL){
			txt_file_opt=0;
		}
		if(strstr(cut_writefile[0],"true")!=NULL){
			txt_file_opt=1;
		}
		break;
		//java에서 File 발견
	case 5:
		txt_check=1;
		char* cut_txtfilename[1];
		char* cut_txtfilename2[1];
		strtok_r(str_read," ",&cut_txtfilename[0]);
		strtok_r(cut_txtfilename[0]," ",&cut_txtfilename2[0]);
```

```
                strcpy(txt_filevar,cut_txtfilename[0]);
                strtok_r(cut_txtfilename2[0],"(",&cut_txtfilename[0]);
                strtok_r(cut_txtfilename[0],")",&cut_txtfilename2[0]);
                strcpy(txt_filename,cut_txtfilename[0]);
                break;
                //java에서 int형 변수 발견
        case 6:
                //초기화된 변수들 따로 저장
                if(strstr(str_read,"=")!=NULL){
                        int conditional_check=0;
                        for(int i=0;i<CONDITIONAL;i++){

if(strstr(str_read,java_conditional[i])!=NULL){
                                        conditional_check=1;
                                        break;
                                }
                        }
                        if(conditional_check)
                                break;
                        char *cut_str7[1];
                        char *cut_str8[1];
                        strtok_r(str_read," ",&cut_str7[0]);
                        strtok_r(cut_str7[0],"=",&cut_str8[0]);
                        strcpy(var_init[var_init_count],cut_str7[0]);
                        var_init_count++;
                        while(strstr(cut_str8[0],",")!=NULL){
                                strtok_r(cut_str8[0]," ",&cut_str7[0]);
                                strtok_r(cut_str7[0],"=",&cut_str8[0]);

strcpy(var_init[var_init_count],cut_str7[0]);
                                var_init_count++;
                        }
                }
                //초기화되지 않은 변수들 따로 저장
                else{
                        char *cut_str9[1];
                        char *cut_str10[1];
                        strtok_r(str_read," ",&cut_str9[0]);
                        strtok_r(cut_str9[0],";",&cut_str10[0]);
                        strcpy(var[var_count],cut_str9[0]);
                        var_count++;
                }
                break;
                //java에서 .write 발견
```

```
                                        case 8:
                                                strcpy(c_func[c_count],"fputs");
                                                c_count++;
                                                //쓰인다고 따로 배열에 저장
                                                if(!save_dw){

sprintf(use_java_func[use],"%s%s",txt_fileDis,java_func[i]);
                                                        sprintf(use_c_func[use],"fputs");
                                                        use++;
                                                        save_dw=1;
                                                }
                                                break;
                                                //java에서 .flush발견
                                        case 9:
                                                strcpy(c_func[c_count],"fflush");
                                                c_count++;
                                                //쓰인다고 따로 배열에 저장
                                                if(!save_ff){

sprintf(use_java_func[use],"%s%s",txt_fileDis,java_func[i]);
                                                        sprintf(use_c_func[use],"fflush");
                                                        use++;
                                                }
                                                break;
                                                //java에서 .close발견
                                        case 10:
                                                strcpy(c_func[c_count],"fclose");
                                                c_count++;
                                                //쓰인다고 따로 배열에 저장
                                                if(!save_fc){

sprintf(use_java_func[use],"%s%s",txt_fileDis,java_func[i]);
                                                        sprintf(use_c_func[use],"fclose");
                                                        use++;
                                                }
                                                break;
                                        }
                                }
                        }
                        continue;
                }
                //class안의 내용을 탐색한다.
                else if (fileType == 2) {
                        //이 아래부터는 main함수가 있는 영역이다.
```

```
if (strstr(str_read, "public class ") != NULL) {
        fileType = 3;
        break;
}
//아직 class안의 내용
//탐색하기
for(int i=0;i<JFUNC;i++){
        if(strstr(str_read,java_func[i])!=NULL){
                switch(i){
                        //java의 printf함수 발견
                        case 0 :
                                strcpy(c_func[c_count],"printf");
                                char *cut_str4[1];
                                char *cut_str5[1];
                                strtok_r(str_read,"(",&cut_str4[0]);
                                strtok_r(cut_str4[0],")",&cut_str5[0]);
                                strcpy(printf_context[printf_count],cut_str4[0]);
                                c_count++;
                                printf_count++;
                                //쓰인다고 따로 배열에 저장해놓기!
                                if(!save_p){
                                        sprintf(use_java_func[use],"%s",java_func[i]);
                                        sprintf(use_c_func[use],"printf");
                                        use++;
                                        save_p=1;
                                }
                                break;
                                //java의 scanf함수 발견
                        case 1 :
                                //약간의 가공과정
                                if(scanf_check==0){
                                        char *cut_str[1];
                                        char *cut_str6[1];
                                        strtok_r(str_read," =",&cut_str[0]);
                                        strtok_r(cut_str[0], " =",&cut_str6[0]);
                                        strcpy(java_func[i],cut_str[0]);
                                        scanf_check=1;
                                }
                                else if(scanf_check==1){
                                        char *cut_str2[1];
                                        char *cut_str3[1];
                                        strtok_r(str_read," ",&cut_str2[0]);
                                        strtok_r(cut_str2[0]," ",&cut_str3[0]);
                                        strtok_r(cut_str3[0],"(",&cut_str2[0]);
```

```
                        strcpy(java_func[i],cut_str3[0]);
                        strcpy(c_func[c_count],"scanf");
                        c_count++;
                        //쓰인다고 따로 배열에 저장해놓기!
                        if(!save_s){

sprintf(use_java_func[use],"%s",java_func[i]);
                                sprintf(use_c_func[use],"scanf");
                                use++;
                                save_s=1;
                        }
                }
                break;
                //java에서 return발견
        case 2:
                strcpy(c_func[c_count],"exit");
                c_count++;
                //쓰인다고 따로 배열에 저장
                if(!save_r){
                        sprintf(use_java_func[use],"%s",java_func[i]);
                        sprintf(use_c_func[use],"exit");
                        use++;
                        save_r = 1;
                }
                break;
                //public찾으면
        case 3:
                if(strstr(str_read,"final")!=NULL){
                        char *cut_str13[1];
                        char *cut_str14[1];
                        strtok_r(str_read," ",&cut_str13[0]);
                        strtok_r(cut_str13[0]," ",&cut_str14[0]);
                        strtok_r(cut_str14[0]," ",&cut_str13[0]);
                        strtok_r(cut_str13[0]," ",&cut_str14[0]);
                        strtok_r(cut_str14[0]," ",&cut_str13[0]);
                        strcpy(use_macro[macro_count],cut_str14[0]);
                        strtok_r(cut_str13[0]," ",&cut_str14[0]);
                        strtok_r(cut_str14[0],";",&cut_str13[0]);
                        strcpy(macro_define[macro_count],cut_str14[0]);
                        macro_count++;
                        break;
                }
                if(!constructor_check){
                        if(strstr(str_read,class_name)!=NULL){
```

```
                        char *cut_str15[1];
                        char *cut_str16[1];
                        strtok_r(str_read," ",&cut_str15[0]);
                        strtok_r(cut_str15[0],"{",&cut_str16[0]);
                        strcpy(use_constructor,cut_str15[0]);
                        break;
                }
                constructor_check=1;
        }
        //이제 사용자정의함수 저장.
        //함수 이름만 저장!
        char *cut_str17[1];
        char *cut_str18[1];
        strtok_r(str_read," ",&cut_str17[0]);
        strtok_r(cut_str17[0]," ",&cut_str18[0]);
        strtok_r(cut_str18[0],"(",&cut_str17[0]);
        strcpy(use_myfunc[myfunc_count],cut_str18[0]);
        myfunc_count++;
        break;
        //java에서 int형 변수 발견
case 6:
        //초기화된 변수들 따로 저장
        if(strstr(str_read,"=")!=NULL){
                int conditional_check=0;
                for(int i=0;i<CONDITIONAL;i++){

if(strstr(str_read,java_conditional[i])!=NULL){
                                conditional_check=1;
                                break;
                        }
                }
                if(conditional_check)
                        break;
                char *cut_str7[1];
                char *cut_str8[1];
                strtok_r(str_read," ",&cut_str7[0]);
                strtok_r(cut_str7[0],"=",&cut_str8[0]);
                strcpy(var_init[var_init_count],cut_str7[0]);
                var_init_count++;
                while(strstr(cut_str8[0],",")!=NULL){
                        strtok_r(cut_str8[0]," ",&cut_str7[0]);
                        strtok_r(cut_str7[0],"=",&cut_str8[0]);

strcpy(var_init[var_init_count],cut_str7[0]);
```

```
                                        var_init_count++;
                                }
                        }
                        //초기화되지 않은 변수들 따로 저장
                        else{
                                char *cut_str9[1];
                                char *cut_str10[1];
                                strtok_r(str_read," ",&cut_str9[0]);
                                strtok_r(cut_str9[0],";",&cut_str10[0]);
                                strcpy(var[var_count],cut_str9[0]);
                                var_count++;
                        }
                        break;
                case 7:
                        if(strstr(str_read,"[]")!=NULL){
                                char *cut_str15[1];
                                char *cut_str16[1];
                                strtok_r(str_read," ",&cut_str15[0]);
                                strtok_r(cut_str15[0],";",&cut_str16[0]);
                                strcpy(use_arr[arr_count],cut_str15[0]);
                                arr_count++;
                                break;
                        }
                }
        }
        continue;
}
if (fileType == 3) {
        //class파일이 만들어져있다.
        //public class가 나올때 까지 계속 읽는다.
        if (strstr(str_read, "public class ") != NULL) {
                fileType = 4;
        }
        if (exp_class) {
                continue;
        }
}
if (fileType == 4) {
        //class파일이 만들어져 있고 public class도 나온 다음의 내용을 읽고 있다.
        if (!first_init) {
                use_class = use;
                //배열들도 다시 복사해준다.
                for(int i=0;i<use;i++){
```

```
                strcpy(use_java_func_class[i], use_java_func[i]);
        }
        for(int i=0;i<use;i++){
                strcpy(use_c_func_class[i], use_c_func[i]);
        }
        //변수들을 모두 초기화한다
        c_count = 0;
        printf_count = 0;
        count = 0;
        use = 0;
        var_count = 0;
        var_init_count = 0;
        first_init = 1;
}
//이제 main함수 영역이다.
//탐색하기
for(int i=0;i<JFUNC;i++){
        if(strstr(str_read,java_func[i])!=NULL){
                switch(i){
                        //java의 printf함수 발견
                        case 0 :
                                strcpy(c_func[c_count],"printf");
                                char *cut_str4[1];
                                char *cut_str5[1];
                                strtok_r(str_read,"(",&cut_str4[0]);
                                strtok_r(cut_str4[0],")",&cut_str5[0]);
                                if(strstr(cut_str5[0],")")!=NULL){
                                        sprintf(cut_str4[0],"%s)",cut_str4[0]);
                                }
                                strcpy(printf_context[printf_count],cut_str4[0]);
                                c_count++;
                                printf_count++;
                                //쓰인다고 따로 배열에 저장
                                if(!save_p){
                                        sprintf(use_java_func[use],"%s",java_func[i]);
                                        sprintf(use_c_func[use],"printf");
                                        use++;
                                        save_p=1;
                                }
                                break;
                                //java의 scanf함수 발견
                        case 1 :
                                //약간의 가공과정
                                if(scanf_check==0){
```

```
                char *cut_str[1];
                char *cut_str6[1];
                strtok_r(str_read," =",&cut_str[0]);
                strtok_r(cut_str[0], " =",&cut_str6[0]);
                strcpy(java_func[i],cut_str[0]);
                scanf_check=1;
        }
        else if(scanf_check==1){
                char *cut_str2[1];
                char *cut_str3[1];
                strtok_r(str_read," ",&cut_str2[0]);
                strtok_r(cut_str2[0]," ",&cut_str3[0]);
                strtok_r(cut_str3[0],"(",&cut_str2[0]);
                strcpy(java_func[i],cut_str3[0]);
                strcpy(c_func[c_count],"scanf");
                c_count++;
                //쓰인다고 따로 배열에 저장
                if(!save_s){

sprintf(use_java_func[use],"%s",java_func[i]);

                        sprintf(use_c_func[use],"scanf");
                        use++;
                        save_s=1;
                }
        }
        break;
        //java에서 return발견
case 2:
        strcpy(c_func[c_count],"exit");
        c_count++;
        //쓰인다고 따로 배열에 저장
        if(!save_r){
                sprintf(use_java_func[use],"%s",java_func[i]);
                sprintf(use_c_func[use],"exit");
                use++;
                save_r = 1;
        }
        break;
        //java에서 int형 변수 발견
case 6:
        //초기화된 변수들 따로 저장
        if(strstr(str_read,"=")!=NULL){
                int conditional_check=0;
                for(int i=0;i<CONDITIONAL;i++){
```

```
if(strstr(str_read,java_conditional[i])!=NULL){
                                                                                conditional_check=1;
                                                                                break;
                                                                        }
                                                                }
                                                                if(conditional_check)
                                                                        break;
                                                                char *cut_str7[1];
                                                                char *cut_str8[1];
                                                                strtok_r(str_read," ",&cut_str7[0]);
                                                                strtok_r(cut_str7[0],"=",&cut_str8[0]);
                                                                strcpy(var_init[var_init_count],cut_str7[0]);
                                                                var_init_count++;
                                                                while(strstr(cut_str8[0],",")!=NULL){
                                                                        strtok_r(cut_str8[0]," ",&cut_str7[0]);
                                                                        strtok_r(cut_str7[0],"=",&cut_str8[0]);

strcpy(var_init[var_init_count],cut_str7[0]);
                                                                        var_init_count++;
                                                                }
                                                        }
                                                        //초기화되지 않은 변수들 따로 저장
                                                        else{
                                                                char *cut_str9[1];
                                                                char *cut_str10[1];
                                                                strtok_r(str_read," ",&cut_str9[0]);
                                                                strtok_r(cut_str9[0],";",&cut_str10[0]);
                                                                strcpy(var[var_count],cut_str9[0]);
                                                                var_count++;
                                                        }
                                                        break;
                                                }
                                        }
                                }
                                continue;
                        }
                }
                fclose(fd);
}

//함수에 따른 헤더를 찾아 c언어에 쓸 헤더를 저장하는 함수이다
void headerfile(char *func){
        int exist=0;
```

```
	for(int j=0;j<headerTable_count;j++){
		if(!strcmp(func,headerTable_func[j])){
			for(int i=0;i<count;i++){
				//헤더파일 저장한 배열에 똑같은 헤더파일이 있다
				if(!strcmp(headerName[i],headerTable_header[j])){
					exist=1;
					break;
				}
			}
			//헤더파일 저장한 배열에 똑같은 헤더파일이 없으므로 저장한다.
			if(exist==0){
				strcpy(headerName[count],headerTable_header[j]);
				count++;
			}
		}
	}
}

//자바를 c로 변환해서 c파일로 저장하는 함수이다.
void convertJavaToC(char* fname, char* c_file_name){
	FILE* c_fd;
	FILE* j_fd;
	char j_str_read[MAX_SIZE];
	int box_count=0;
	char class_name_new[MAX_SIZE];
	//c파일 만들기
	if((c_fd=fopen(c_file_name,"w+"))==NULL){
		fprintf(stderr, "fopen error for %s\n",c_file_name);
		exit(1);
	}
	//헤더파일 입력하기
	for(int i=0;i<count;i++){
		fputs(headerName[i],c_fd);
		fputs("\n",c_fd);
	}
	//java파일 열기
	if((j_fd=fopen(fname,"r"))==NULL){
		fprintf(stderr, "fopen error for %s\n",fname);
		exit(1);
	}
	int pf_check=0;
	int var_check=0;

	//java파일 읽기
```

```
while(1){
	fgets(j_str_read,sizeof(j_str_read),j_fd);
	if(feof(j_fd))
		break;
	if(fileType==4){
		if(strstr(j_str_read,"public class ")!=NULL){
			fileType=5;
		}
		else{
			continue;
		}
	}
	if(strstr(j_str_read,"main")!=NULL){
		if(strstr(j_str_read,"{")!=NULL){
			box_count++;
			fputs("int main(void){\n",c_fd);
		}
		else{
			b_check=1;
			fputs("int main(void)\n",c_fd);
		}
		continue;
	}
	if(no_box_check2){
		box_count--;
		no_box_check2=0;
	}
	if(no_box_check){
		if(strstr(j_str_read,"{")==NULL){
			box_count++;
			no_box_check=0;
			no_box_check2=1;
		}
		else
			no_box_check=0;
	}
	if(!strcmp(j_str_read,"\n"))
		fputs(j_str_read,c_fd);
	//앞의 공백 자르기
	EraseFrontSpace(j_str_read);
	if(strcmp(j_str_read,"}\n")){
		if(strstr(j_str_read,"}")!=NULL){
			char *bb_remove[1];
			strtok_r(j_str_read,"}",&bb_remove[0]);
```

```
                bc_check=1;
                box_count--;
        }
}
if(bc_check){
        if(!strcmp(j_str_read,"}\n"))
                bc_check=0;
        else{
                addTab(box_count,c_fd);
                fputs("}\n",c_fd);
                bc_check=0;
        }
}
if(fileType==3){
        if(strstr(j_str_read,"{")!=NULL)
                box_count++;
        if(strstr(j_str_read,"return")!=NULL){
                addTab(box_count,c_fd);
                fputs(j_str_read,c_fd);
                if(strstr(j_str_read,"}")!=NULL){
                        box_count--;
                }
                continue;
        }
        if(!strcmp(j_str_read,"}\n")){
                box_count--;
                if(box_count==0)
                        break;
                addTab(box_count,c_fd);
                fputs(j_str_read,c_fd);
                continue;
        }
        if(strstr(j_str_read,"}")!=NULL){
                box_count--;
                if(box_count==0)
                        break;
                addTab(box_count,c_fd);
                fputs(j_str_read,c_fd);
                continue;
        }
        //printf, scanf등등 함수 탐색
        int f_check=0;
        for(int i=0;i<use;i++){
                if(strstr(j_str_read,use_java_func[i])!=NULL){
```

```
                    if(strstr(use_java_func[i],"printf")!=NULL){
                            addTab(box_count,c_fd);
                            fputs(use_c_func[i],c_fd);
                            fputs("(",c_fd);
                            fputs(printf_context[pf_check],c_fd);
                            pf_check++;
                            fputs(");\n",c_fd);
                            f_check=1;
                    }
                    else if(strstr(use_java_func[i],"return")!=NULL){
                            addTab(box_count,c_fd);
                            fputs(use_c_func[i],c_fd);
                            fputs("(0);\n",c_fd);
                            f_check=1;
                    }
                    else{
                            addTab(box_count,c_fd);
                            fputs(use_c_func[i],c_fd);
                            fputs("(\"%d\",&",c_fd);
                            fputs(var[var_check],c_fd);
                            var_check++;
                            fputs(");\n",c_fd);
                            f_check=1;
                    }
            }
    }
    if(f_check)
            continue;
    if(strstr(j_str_read,"public ")!=NULL){
            if(strstr(j_str_read,"static final int")!=NULL){
                    for(int i=0;i<macro_count;i++){
                            addTab(box_count,c_fd);
                            fputs("#define ",c_fd);
                            fputs(use_macro[i],c_fd);
                            fputs(" ",c_fd);
                            fputs(macro_define[i],c_fd);
                            fputs("\n",c_fd);
                    }
                    continue;
            }
            if(constructor_check){
                    if(strstr(j_str_read,use_constructor)!=NULL){
                            addTab(box_count-1,c_fd);
                            constructor_check=1;
```

```
                        fputs("void ",c_fd);
                        fputs(use_constructor,c_fd);
                        fputs("{\n",c_fd);
                }
                constructor_check=0;
        }
        for(int i=0;i<myfunc_count;i++){
                if(strstr(j_str_read,use_myfunc[i])!=NULL){
                        char *cut_myfunc_str[0];
                        strtok_r(j_str_read," ",&cut_myfunc_str[0]);
                        fputs(cut_myfunc_str[0],c_fd);
                        continue;
                }
        }
}
//변수 탐색
if(strstr(j_str_read,"int ")!=NULL){
        int con_check=0;
        for(int i=0;i<CONDITIONAL;i++){
                if(strstr(j_str_read,java_conditional[i])!=NULL){
                        con_check=1;
                        break;
                }
        }
        if(!con_check){
                addTab(box_count,c_fd);
                fputs(j_str_read,c_fd);
                continue;
        }
}
int v_check=0;
for(int i=0;i<var_count;i++){
        if(strstr(j_str_read,var[i])!=NULL){
                int con_check2=0;
                for(int i=0;i<CONDITIONAL;i++){
                        if(strstr(j_str_read,java_conditional[i])!=NULL){
                                con_check2=1;
                                break;
                        }
                }
                if(!con_check2){
                        addTab(box_count,c_fd);
                        fputs(j_str_read,c_fd);
                        v_check=1;
```

```
                }
                break;
        }
}
if(v_check)
        continue;
if(strstr(j_str_read,"int[] ")!=NULL){
        addTab(box_count,c_fd);
        fputs("int *",c_fd);
        for(int i=0;i<arr_count;i++){
                fputs(use_arr[i],c_fd);
                fputs(" = NULL;\n",c_fd);
        }
        continue;
}
//여기는 stack = new int[STACK_SIZE]; 이게 걸림.
for(int i=0;i<arr_count;i++){
        if(strstr(j_str_read,use_arr[i])!=NULL){
                if(strstr(j_str_read,"new ")!=NULL){
                        addTab(box_count,c_fd);
                        fputs(use_arr[i],c_fd);
                        char *cut_arr_size[1];
                        char *cut_arr_size2[1];
                        strtok_r(j_str_read,"[",&cut_arr_size[0]);
                        strtok_r(cut_arr_size[0],"]",&cut_arr_size2[0]);
                        fputs(" =(int*)malloc(sizeof(int)*",c_fd);
                        fputs(cut_arr_size[0],c_fd);
                        fputs(");\n",c_fd);
                        arr_check=1;
                        break;
                }
        }
}
if(arr_check){
        arr_check=0;
        continue;
}
//for문, if문등 탐색
int c_check=0;
for(int i=0;i<CONDITIONAL;i++){
        if(strstr(j_str_read,java_conditional[i])!=NULL){
                if(strstr(j_str_read,"{")==NULL){
                        no_box_check=1;
                        b_check=1;
```

```
                        }
                        addTab(box_count-1,c_fd);
                        fputs(j_str_read,c_fd);
                        c_check=1;
                }
        }
        for(int i=0;i<var_init_count;i++){
                if(strstr(j_str_read,var_init[i])!=NULL){
                        addTab(box_count,c_fd);
                        fputs(j_str_read,c_fd);
                }
        }
        if(strstr(j_str_read,"}")!=NULL){
                box_count--;
                if(box_count==0)
                        break;
                addTab(box_count,c_fd);
                fputs(j_str_read,c_fd);
        }
        if(c_check)
                continue;
}
else{
        if(fileType==5){
                if(strstr(j_str_read,"}")!=NULL){
                        if(arr_check){
                                for(int i=0;i<arr_count;i++){
                                        fputs("\n",c_fd);
                                        addTab(box_count,c_fd);
                                        fputs("free(",c_fd);
                                        fputs(use_arr[i],c_fd);
                                        fputs(");\n",c_fd);
                                }
                        }
                        box_count--;
                        fputs(j_str_read,c_fd);
                        if(box_count==0)
                                break;
                }
        }
        //변수 탐색
        if(strstr(j_str_read,"int ")!=NULL){
                int con_check=0;
                for(int i=0;i<CONDITIONAL;i++){
```

```
                        if(strstr(j_str_read,java_conditional[i])!=NULL){
                                con_check=1;
                                break;
                        }
                }
                if(!con_check){
                        addTab(box_count,c_fd);
                        fputs(j_str_read,c_fd);
                        continue;
                }
        }
        //printf, scanf등등 함수 탐색
        int f_check=0;
        for(int i=0;i<use;i++){
                if(strstr(j_str_read,use_java_func[i])!=NULL){
                        if(strstr(use_java_func[i],"printf")!=NULL){
                                addTab(box_count,c_fd);
                                fputs(use_c_func[i],c_fd);
                                fputs("(",c_fd);
                                char* cut_myf2[1];
                                char* cut_myf3[1];
                                if(cl_check==1){

if(strstr(printf_context[pf_check],class_name_new)!=NULL){
                                                strtok_r(printf_context[pf_check],",",
&cut_myf2[0]);
                                                strtok_r(cut_myf2[0],".",&cut_myf3[0]);
                                                fputs(printf_context[pf_check],c_fd);
                                                fputs(", ",c_fd);
                                        }
                                        else
                                                fputs(printf_context[pf_check],c_fd);
                                }
                                else
                                        fputs(printf_context[pf_check],c_fd);
                                if(cl_check==1)
                                        fputs(cut_myf3[0],c_fd);
                                pf_check++;
                                fputs(");\n",c_fd);
                                f_check=1;
                                break;
                        }
                        else if(strstr(use_java_func[i],"return")!=NULL){
                                addTab(box_count,c_fd);
```

```
            fputs(use_c_func[i],c_fd);
            fputs("(0);\n",c_fd);
            f_check=1;
            break;
    }
    else if(strstr(use_java_func[i],"FileWriter ")!=NULL){
            break;
    }
    else if(strstr(use_java_func[i],".write")!=NULL){
            addTab(box_count,c_fd);
            char* cut_w_context[1];
            char* cut_w_context2[1];
            strtok_r(j_str_read,"(",&cut_w_context[0]);
            strtok_r(cut_w_context[0],")",&cut_w_context2[0]);
            fputs(use_c_func[i],c_fd);
            fputs("(",c_fd);
            fputs(cut_w_context[0],c_fd);
            fputs(",",c_fd);
            fputs(txt_fileDis,c_fd);
            fputs(");\n",c_fd);
            f_check=1;
            break;
    }
    else if(strstr(use_java_func[i],".flush")!=NULL){
            addTab(box_count,c_fd);
            fputs(use_c_func[i],c_fd);
            fputs("(",c_fd);
            fputs(txt_fileDis,c_fd);
            fputs(");\n",c_fd);
            f_check=1;
            break;
    }
    else if(strstr(use_java_func[i],".close")!=NULL){
            addTab(box_count,c_fd);
            fputs(use_c_func[i],c_fd);
            fputs("(",c_fd);
            fputs(txt_fileDis,c_fd);
            fputs(");\n",c_fd);
            f_check=1;
            break;
    }
    else{
            addTab(box_count,c_fd);
            fputs(use_c_func[i],c_fd);
```

```
                                fputs("(\"%d\",&",c_fd);
                                fputs(var[var_check],c_fd);
                                var_check++;
                                fputs(");\n",c_fd);
                                f_check=1;
                                break;
                        }
                }
        }
        if(f_check)
                continue;
        //for문, if문등 탐색
        int c_check=0;
        for(int i=0;i<CONDITIONAL;i++){
                if(strstr(j_str_read,java_conditional[i])!=NULL){
                        if(strstr(j_str_read,"{")==NULL){
                                no_box_check=1;
                                b_check=1;
                        }
                        addTab(box_count,c_fd);
                        if(strstr(j_str_read, "null")!=NULL){
                                char *cut_null1[1];
                                char *cut_null2[1];
                                strtok_r(j_str_read," ",&cut_null1[0]);
                                strtok_r(cut_null1[0], " ",&cut_null2[0]);
                                sprintf(j_str_read,"%s %s NULL)\n",j_str_read,cut_null1[0]);
                        }
                        fputs(j_str_read,c_fd);
                        c_check=1;
                }
        }
        if(strstr(j_str_read,"{")!=NULL){
                if(!b_check){
                        box_count++;
                }
        }
        for(int i=0;i<var_init_count;i++){
                if(strstr(j_str_read,var_init[i])!=NULL){
                        addTab(box_count,c_fd);
                        fputs(j_str_read,c_fd);
                        break;
                }
        }
        if(strstr(j_str_read,"}")!=NULL){
```

```
                box_count--;
                if(box_count==0)
                        break;
                addTab(box_count,c_fd);
                fputs(j_str_read,c_fd);
        }
        if(c_check)
                continue;
}
if(fileType==5){
        if(strstr(j_str_read,"}")!=NULL){
                addTab(box_count,c_fd);
                fputs(j_str_read,c_fd);
                box_count--;
                if(box_count==0)
                        break;
        }
        if(!cl_check){
                if(strstr(j_str_read,class_name)!=NULL){
                        char* class_cons[1];
                        char* class_cons2[1];
                        strtok_r(j_str_read," ",&class_cons[0]);
                        strtok_r(class_cons[0]," =",&class_cons2[0]);
                        sprintf(class_name_new,"%s.",class_cons[0]);//class_name_new = st.
                        fputs("\n", c_fd);
                        addTab(box_count,c_fd);
                        fputs(use_constructor,c_fd);
                        fputs(";\n",c_fd);
                        cl_check=1;
                }
        }
}
if(txt_check){
        if(strstr(j_str_read,"/*")!=NULL){
                addTab(box_count,c_fd);
                if(strstr(j_str_read,"true")!=NULL){
                        char* remark[1];
                        char* remark2[1];
                        strtok_r(j_str_read,"t",&remark[0]);
                        fputs(j_str_read,c_fd);
                        fputs("\"a\"",c_fd);
                        strtok_r(remark[0],"e",&remark2[0]);
                        fputs(remark2[0],c_fd);
                }
```

```
		else if(strstr(j_str_read,"false")!=NULL){
			char* remark3[1];
			char* remark4[1];
			strtok_r(j_str_read,"f",&remark3[0]);
			fputs(j_str_read,c_fd);
			fputs("\"w\"",c_fd);
			strtok_r(remark3[0],"e",&remark4[0]);
			fputs(remark4[0],c_fd);
		}
		else
			fputs(j_str_read,c_fd);
		continue;
	}
	if(strstr(j_str_read,"File ")!=NULL){
		addTab(box_count,c_fd);
		fputs("char* ",c_fd);
		fputs(txt_filevar,c_fd);
		fputs(" = ",c_fd);
		fputs(txt_filename,c_fd);
		fputs(";\n",c_fd);
		continue;
	}
	if(strstr(j_str_read,"FileWriter ")!=NULL){
		char fopen_opt[10];
		if(txt_file_opt){
			strcpy(fopen_opt,"\"a\"");
		}
		else{
			strcpy(fopen_opt,"\"w\"");
		}
		addTab(box_count,c_fd);
		fputs("FILE* ",c_fd);
		fputs(txt_fileDis,c_fd);
		fputs(" = fopen(",c_fd);
		fputs(txt_filevar,c_fd);
		fputs(", ",c_fd);
		fputs(fopen_opt,c_fd);
		fputs(");\n",c_fd);
		addTab(box_count,c_fd);
		fputs("if(",c_fd);
		fputs(txt_fileDis,c_fd);
		fputs(" == NULL){\n",c_fd);
		addTab(box_count+1,c_fd);
		fputs("fprintf(stderr, \"open error for %s\\n\", ",c_fd);
```

```
                                fputs(txt_filevar,c_fd);
                                fputs(");\n",c_fd);
                                addTab(box_count+1,c_fd);
                                fputs("exit(1);\n",c_fd);
                                addTab(box_count,c_fd);
                                fputs("}\n",c_fd);
                                continue;
                        }
                }
                if(cl_check==1){
                        if(strstr(j_str_read,class_name_new)!=NULL){
                                addTab(box_count,c_fd);
                                char* cut_myf[1];
                                strtok_r(j_str_read, ".", &cut_myf[0]);
                                fputs(cut_myf[0],c_fd);
                                continue;
                        }
                }
                if(strstr(j_str_read,"{")!=NULL){
                        if(b_check){
                                b_check=0;
                                addTab(box_count,c_fd);
                                box_count++;
                                fputs("{\n",c_fd);
                        }
                }
        }
        fclose(j_fd);
        fclose(c_fd);
}

//p옵션을 실행할 때 수행되는 함수이다.
//어떤 java의 함수를 c언어의 함수로 바꿨는지를 보여준다.
void optionP(int p_flag){
        int count=1;
        if(p_flag){
                if(exp_class){
                        int u_class=use_class;
                        int y_check=0;
                        for(int i=0;i<use;i++){
                                for(int j=0;j<u_class;j++){
                                        if(!strcmp(use_java_func_class[j],use_java_func[i])){
                                                y_check=1;
                                        }
```

```
                        }
                        if(!y_check){
                                strcpy(use_java_func_class[use_class],use_java_func[i]);
                                strcpy(use_c_func_class[use_class],use_c_func[i]);
                                use_class++;
                                y_check=0;
                        }
                }
                for(int i=0;i<use_class;i++){
                        if(strstr(use_java_func_class[i],"return")!=NULL)
                                continue;
                        printf("%d %s() -> %s()\n",count,use_java_func_class[i],use_c_func_class[i]);
                        count++;
                }
        }
        else{
                for(int i=0;i<use;i++){
                        if(strstr(use_java_func[i],"return")!=NULL)
                                continue;
                        printf("%d %s() -> %s()\n",count,use_java_func[i],use_c_func[i]);
                        count++;
                }
        }
    }
}

//l옵션을 실행할 때 수행되는 함수이다.
//java파일과 c파일의 라인 수를 출력해준다.
void optionL(int l_flag, char*cPath, char*c_fname, char*jPath, char*j_fname){
        char j_str_read[2048];
        FILE *j_fd;
        jLine=0;
        char c_str_read[2048];
        FILE *c_fd;
        cLine=0;
        if(l_flag){
                if((j_fd=fopen(jPath, "r"))==NULL){
                        fprintf(stderr, "fopen error for %s\n",jPath);
                        exit(1);
                }
                else{
                        while(1){
                                fgets(j_str_read,2048,j_fd);
                                if(feof(j_fd))
```

```
                        break;
                jLine++;
        }
}
printf("%s line number is %d lines\n",j_fname, jLine);
fclose(j_fd);
if(exp_class){
        char class_str_read[2048];
        FILE *class_fd;
        char class_filename[MAX_SIZE];
        char class_n[MAX_SIZE];
        char *cut[1];
        strcpy(class_n,use_constructor);
        strtok_r(class_n,"(",&cut[0]);
        sprintf(class_filename,"%s.c",class_n);
        int class_line=0;
        if((class_fd=fopen(class_filename, "r"))==NULL){
                fprintf(stderr, "fopen error for %s\n",class_filename);
                exit(1);
        }
        else{
                while(1){
                        fgets(class_str_read,2048,class_fd);
                        if(feof(class_fd))
                                break;
                        class_line++;
                }
        }
        printf("%s line number is %d lines\n",class_filename, class_line);
        fclose(class_fd);
}
if((c_fd=fopen(cPath, "r"))==NULL){
        fprintf(stderr, "fopen error for %s\n",cPath);
        exit(1);
}
else{
        while(1){
                fgets(c_str_read,2048,c_fd);
                if(feof(c_fd))
                        break;
                cLine++;
        }
}
printf("%s line number is %d lines\n",c_fname, cLine);
```

```
                fclose(c_fd);
        }
}

//j옵션을 실행할 때 수행되는 함수이다.
//java파일의 내용을 보여준다.
void optionJ(int j_flag, char* path,char* fname){
        char j_str_read[2048];
        FILE *fd;
        jLine=0;
        if(j_flag){
                if((fd=fopen(path, "r"))==NULL){
                        fprintf(stderr, "fopen error for %s\n",path);
                        exit(1);
                }
                else{
                        while(1){
                                fgets(j_str_read,2048,fd);
                                if(feof(fd))
                                        break;
                                printf("%d %s",jLine+1,j_str_read);
                                jLine++;
                        }
                }
                fclose(fd);
        }
}

//c옵션을 실행할 때 수행되는 함수이다.
//c언어로 변환한 내용을 보여준다.
void optionC(int c_flag, char* path,char* fname){
        char c_str_read[2048];
        FILE *fd;
        cLine=0;
        if(c_flag){
                if(exp_class){
                        char class_str_read[2048];
                        FILE *class_fd;
                        int class_line=0;
                        char class_filename[MAX_SIZE];
                        char class_n[MAX_SIZE];
                        char *cut[1];
                        strcpy(class_n,use_constructor);
                        strtok_r(class_n,"(",&cut[0]);
```

```
                sprintf(class_filename,"%s.c",class_n);
                printf("%s\n",class_filename);
                if((class_fd=fopen(class_filename, "r"))==NULL){
                        fprintf(stderr, "fopen error for %s\n",class_filename);
                        exit(1);
                }
                else{
                        while(1){
                                fgets(class_str_read,2048,class_fd);
                                if(feof(class_fd))
                                        break;
                                printf("%d %s",class_line+1, class_str_read);
                                class_line++;
                        }
                }
                fclose(class_fd);
                printf("\n%s\n",fname);
        }
        if((fd=fopen(path, "r"))==NULL){
                fprintf(stderr, "fopen error for %s\n",path);
                exit(1);
        }
        else{
                while(1){
                        fgets(c_str_read,2048,fd);
                        if(feof(fd))
                                break;
                        printf("%d %s",cLine+1,c_str_read);
                        cLine++;
                }
        }
        fclose(fd);
    }
}

//f옵션을 실행할 때 수행되는 함수이다.
void optionF(int f_flag, char*cPath, char*c_fname, char*jPath, char*j_fname){
    struct stat cfd;
    struct stat jfd;
    if(f_flag){
        if(stat(jPath,&jfd)<0)
                fprintf(stderr, "stat error for %s\n",jPath);
        else
                printf("%s file size is %ld bytes\n",j_fname, jfd.st_size);
```

```
                if(exp_class){
                        struct stat classfd;
                        char class_filename[MAX_SIZE];
                        char class_n[MAX_SIZE];
                        char *cut[1];
                        strcpy(class_n,use_constructor);
                        strtok_r(class_n,"(",&cut[0]);
                        sprintf(class_filename,"%s.c",class_n);
                        if(stat(class_filename, &classfd)<0)
                                fprintf(stderr, "stat error for %s\n",jPath);
                        else
                                printf("%s file size is %ld bytes\n",class_filename, classfd.st_size);
                }
                if(stat(cPath,&cfd)<0)
                        fprintf(stderr, "stat error for %s\n",cPath);
                else
                        printf("%s file size is %ld bytes\n",c_fname, cfd.st_size);
        }
}

//r옵션을 실행할 때 수행되는 함수이다.
//변환과정을 보여준다.
void optionR(int r_flag, char *cPath, char* c_name, char* jPath, char*j_name){
        int java_line=0;
        int java_rep=1;
        int c_line=0;
        int c_rep=1;
        int class_line=0;
        int class_rep=1;
        int end_check_j=0;
        int end_check_c=0;
        int end_check_class=0;
        FILE *fd;
        char j_str_read[MAX_SIZE];
        char c_str_read[MAX_SIZE];
        pid_t pid;
        int status;
        int level1_j=0;
        int level1_c=0;
        int level1_class=0;
        int cls_check1=0;
        int cls_check2=0;
        char class_str_read[2048];
        FILE *class_fd;
```

```
char class_filename[MAX_SIZE];
if(r_flag){
        pid=fork();//fork()발생
        if(pid<0){
                printf("fork error\n");
                exit(1);
        }
        sleep(1);
        if(pid==0){//자식 프로세스에서 하는 일
                while(1){
                        java_line=0;
                        c_line=0;
                        class_line=0;
                        system("clear");
                        printf("--------\n");
                        printf("%s\n",j_name);
                        printf("--------\n");
                        fd=fopen(j_name, "r");

                        //자바파일을 보여준다
                        while(1){
                                if(feof(fd)){
                                        end_check_j=1;
                                        break;
                                }
                                java_line++;
                                fgets(j_str_read,sizeof(j_str_read),fd);
                                if(cls_check1==0 && cls_check2==0){
                                        if(strstr(j_str_read,"public class ")!=NULL){
                                                cls_check2=1;
                                        }
                                        if(strstr(j_str_read, "import ")!=NULL){
                                                cls_check2=1;
                                        }
                                        else
                                                cls_check1=1;
                                }
                                if(cls_check1){
                                        if(strstr(j_str_read,"public class ")!=NULL){
                                                cls_check2=1;
                                        }
                                }
                                printf("%d %s",java_line,j_str_read);
                                sleep(0.9);
```

```
		if(java_line==java_rep){
			fclose(fd);
			break;
		}
	}
	if(!strcmp(j_str_read,"\n")){
		level1_j=1;
	}
	//클래스파일의 변환내용을 보여준다
	if(cls_check1){
		char class_n[MAX_SIZE];
		char *cut[1];
		strcpy(class_n,use_constructor);
		strtok_r(class_n,"(",&cut[0]);
		sprintf(class_filename,"%s.c",class_n);
		printf("--------\n");
		printf("%s\n",class_filename);
		printf("--------\n");
		class_fd=fopen(class_filename, "r");
		while(1){
			if(feof(class_fd)){
				end_check_class=1;
				break;
			}
			class_line++;
			fgets(class_str_read,sizeof(class_str_read),class_fd);
			printf("%d %s",class_line,class_str_read);
			sleep(0.9);
			if(class_line==class_rep){
				fclose(class_fd);
				break;
			}
		}
		if(!strcmp(class_str_read,"\n")){
			level1_class=1;
		}
	}
	//c언어로 변환한 파일의 변환내용을 보여준다.
	if(cls_check2){
		printf("--------\n");
		printf("%s\n",c_name);
		printf("--------\n");
		fd=fopen(c_name, "r");
		while(1){
```

```
                if(feof(fd)){
                        end_check_c=1;
                        break;
                }
                c_line++;
                fgets(c_str_read,sizeof(c_str_read),fd);
                printf("%d  %s",c_line,c_str_read);
                sleep(0.9);
                if(c_line==c_rep){
                        fclose(fd);
                        break;
                }
        }
}
if(cls_check1){
        if(!strcmp(class_str_read,"\n")){
                level1_class=1;
        }
        if(exp_class && (end_check_class==0)){
                class_rep++;
        }
        if(!end_check_class){
                if(level1_j!=1&&level1_class==1) {
                        class_rep--;
                }
                else if(level1_j==1 && level1_class!=1){
                        java_rep--;
                }
                else if(level1_j==1 && level1_class==1){
                        level1_j=0;
                        level1_class=0;
                }
        }
}
if(cls_check2){
        if(end_check_j!=1&&end_check_c==1) {
                c_rep--;
        }
        else if(end_check_j==1 && end_check_c!=1){
                java_rep--;
        }
        else if(end_check_j==1 && end_check_c==1){
                break;
        }
```

```
                                        if(!strcmp(c_str_read,"\n")){
                                                level1_c=1;
                                        }
                                        if(level1_j!=1&&level1_c==1) {
                                                c_rep--;
                                        }
                                        else if(level1_j==1 && level1_c!=1){
                                                java_rep--;
                                        }
                                        else if(level1_j==1 && level1_c==1){
                                                level1_j=0;
                                                level1_c=0;
                                        }
                                        c_rep++;
                                }
                                java_rep++;
                                sleep(1);
                        }
                        exit(0);
                        //자식프로세스 종료
                }
                else{
                        wait(&status);
                }
        }
}

void EraseFrontSpace(char *j_str_read){
        char tabString[MAX_SIZE];
        strcpy(tabString,j_str_read);
        int j=0;
        for(int i=0;tabString[i];i++){
                if(tabString[i]=='\t')
                        continue;
                if(tabString[i]=='\n'){
                        j_str_read[j]='\n';
                        j_str_read[j+1]='\0';
                        break;
                }
                j_str_read[j]=tabString[i];
                j++;
        }
}
```