컴파일러 구성 과제#2

1605020 박소현

1642041 이경연

목차

[1. Code 3](#_Toc7146255)

[2.No error in the input data file 17](#_Toc7146256)

[3.With error in the input data file 20](#_Toc7146257)

[4.Data file given from professor 23](#_Toc7146258)

5.Screenshots………………………………………………………………………………………………………………………………………29

1. Code
   1. scanner\_lab.l

|  |
| --- |
| %{  /\*  \* scanner\_lab.l - lexical analyzer  \*렉스를 통해 토큰을 쪼개고 종류를 정한다.  \* Programmer - 이경연, 박소현  \*  \* date - 19.04.26  \*  \*  \*  \*/  #include <stdio.h>  #include <stdlib.h>  #include "tn.h"  #include "glob.h"  %}  %%  "const" return(TCONST);  "else" return(TELSE);  "if" return(TIF);  "int" return(TINT);  "return" return(TRETURN);  "void" return(TVOID);  "while" return(TWHILE);  "+" return(TPLUS);  "-" return(TMINUS);  "\*" return(TMUL);  "/" return(TDIV);  "%" return(TPERCENT);  "=" return(TASSIGN);  "+=" return(TADDASSIGN);  "-=" return(TSUBASSIGN);  "\*=" return(TMULASSIGN);  "/=" return(TDIVASSIGN);  "%=" return(TMODASSIGN);  "!" return(TNOT);  "&&" return(TAND);  "||" return(TOR);  "==" return(TEQUAL);  "!=" return(TNOTEQU);  "<" return(TLESS);  ">" return(TGREAT);  "<=" return(TEQLESS);  ">=" return(TEQGREAT);  "++" return(TINC);  "--" return(TDEC);  "(" return(TSMALLBRACE\_L);  ")" return(TSMALLBRACE\_R);  "[" return(TBIGBRACE\_L);  "]" return(TBIGBRACE\_R);  "{" return(TMIDBRACE\_L);  "}" return(TMIDBRACE\_R);  [,] return(TCOLON);  [;] return(TSEMICOLON);  [\n] return(TNEWLINE);  [ \t] ;  "/\*"([^\*]|\\*+[^\*/])\*\\*\*"\*/" ;  "//".\* ;  0[0-7]+ return(TOCTA);  0(x|X)[0-9A-Fa-f]+ return(THEXA);  0|[1-9][0-9]\* return(TNUMBER);  [+-]?([0-9]\*\.[0-9]+)([eE][-+]?[0-9]+)? return(TREAL);  [A-Za-z\_][A-Za-z0-9\_][A-Za-z0-9\_][A-Za-z0-9\_][A-Za-z0-9\_][A-Za-z0-9\_][A-Za-z0-9\_][A-Za-z0-9\_][A-Za-z0-9\_]+ return(TLONGERR);  [A-Za-z\_][A-Za-z0-9\_]\* return(TIDENT);  \"(([^\"]|\\\")\*[^\\])?\" return(TSTRING);  \'[^']['] return(TCHAR);  [1-9][0-9]\*[A-Za-z0-9\_]+ return(TERROR\_DIGIT\_START);  [A-Za-z\_]+[^A-Za-z0-9\_ \n\t(){}[]]+[A-Za-z0-9\_]\*? return(TERROR\_ILL\_CHAR\_1);  [^A-Za-z0-9\_ \n\t(){}[]]+[A-Za-z0-9\_]\*? return(TERROR\_ILL\_CHAR\_2);  . return(TERROR);  %%  int yywrap()  {  return 1;  } |

* 1. main.c

|  |
| --- |
| /\*  main.c-렉스로 부터 받은 토큰들을 케이스에 따라 다르게 출력하고, 식별자는 해시테이블 작업을 수행한다.  programmer-이경연, 박소현  date-19.04.26  \*/  #include <stdio.h>  #include <stdlib.h>  #include "tn.h"  #include "glob.h"  //식별자로 인식된 종류의 토큰을 처리하는 함수이다.  //식별자들은 ST을 거쳐 Hstable에 저장한다  void iden\_handler(char \* yt) {  int i;  yytext\_for\_ht = yt;  input\_char = yt[indexyy];  leng = strlen(yt);  ReadID();  err = noerror;  if (err == noerror) {  if (nextfree == STsize) {  err = overst;  PrintError(err);  }  ST[nextfree++] = '\0';  ComputeHS(nextid, nextfree); //읽어온 값들에 대해 hash code를 계산  LookupHS(nextid, hashcode); //계산된 hash code값이 hash table에 이미 있는지 확인  //hash table에 해당 hash code값이 없을 경우  if (!found&&ST[nextid] != NULL) {  //변수의 길이가 너무 길지 않을 경우  if ((nextfree - nextid - 1) <= 10) {  ADDHT(hashcode); //해당 hash code를 hash table에 저장해준다.  }  //변수의 길이가 너무 길 경우  else if ((nextfree - nextid - 1) > 10) {  nextfree = nextid; //string table에 변수가 저장되지 않도록 nextid값을 다시 nextfree에 넣어준다.  }  }  //hash table에 해당 hash code값이 있는 경우  else if (found) {  nextfree = nextid; //string table에 변수가 저장되지 않도록 nextid값을 다시 nextfree에 넣어준다.  }  }  }  //lex로부터 나온 토큰의 종류에 따라 출력을 다르게한다.  //에러가 날 경우 PrintError.c를 통해 에러를 구분해 출력한다.  void main()  {  PrintHeading();  int linenum = 1;  int base;  int value;  char \* yt;  enum tokentypes tn; // token number  extern char \*yytext;  while ((tn = yylex()) != TEOF) {  //printf("Line number= %d\t", linenum);  PrintError(noerror);  switch (tn) {  //예약어  case TCONST: printf("%d TCONST %22s\n", linenum, yytext); break;  case TELSE: printf("%d TELSE %22s\n", linenum, yytext); break;  case TIF: printf("%d TIF %22s\n", linenum, yytext); break;  case TINT: printf("%d TINT %22s\n", linenum, yytext); break;  case TRETURN: printf("%d TRETURN %22s\n", linenum, yytext); break;  case TVOID: printf("%d TVOID %22s\n", linenum, yytext); break;  case TWHILE: printf("%d TWHILE %22s\n", linenum, yytext); break;  //사칙연산자  case TPLUS: printf("%d TPLUS %22s\n", linenum, yytext); break;  case TMINUS: printf("%d TMINUX %22s\n", linenum, yytext); break;  case TMUL: printf("%d TMUL %22s\n", linenum, yytext); break;  case TDIV: printf("%d TDIV %22s\n", linenum, yytext); break;  case TPERCENT: printf("%d TPERCENT %22s\n", linenum, yytext); break;  //배정연산자  case TADDASSIGN: printf("%d TADDASSIGN %16s\n", linenum, yytext); break;  case TSUBASSIGN: printf("%d TSUBASSIGN %16s\n", linenum, yytext); break;  case TMULASSIGN: printf("%d TMULASSIGN %16s\n", linenum, yytext); break;  case TDIVASSIGN: printf("%d TDIVASSIGN %16s\n", linenum, yytext); break;  case TMODASSIGN: printf("%d TMODASSIGN %16s\n", linenum, yytext); break;  case TASSIGN: printf("%d TASSIGN %16s\n", linenum, yytext); break;  //논리연산자  case TNOT: printf("%d TNOT %21s\n", linenum, yytext); break;  case TAND: printf("%d TAND %21s\n", linenum, yytext); break;  case TOR: printf("%d TOR %21s\n", linenum, yytext); break;  //관계연산자  case TEQUAL: printf("%d TEQUAL %19s\n", linenum, yytext); break;  case TNOTEQU: printf("%d TNOTEQU %19s\n", linenum, yytext); break;  case TLESS: printf("%d TLESS %19s\n", linenum, yytext); break;  case TGREAT: printf("%d TGREAT %19s\n", linenum, yytext); break;  case TEQLESS: printf("%d TEQLESS %19s\n", linenum, yytext); break;  case TEQGREAT: printf("%d TEQGREAT %19s\n", linenum, yytext); break;  //증감연산자  case TINC: printf("%d TINC %21s\n", linenum, yytext); break;  case TDEC: printf("%d TDEC %21s\n", linenum, yytext); break;  //Identifier  case TIDENT:  printf("%d TIDENT %18s %d\n", linenum, yytext, nextfree);  yt = yytext; iden\_handler(yt); indexyy = 0; break;  //스트링과 캐릭터  case TSTRING:printf("%d TSTRING %18s\n", linenum, yytext); break;  case TCHAR:printf("%d TCHAR %18s\n", linenum, yytext); break;  //특수기호  case TSMALLBRACE\_L: printf("%d TSAMLL\_L %18s\n", linenum, yytext); break;  case TSMALLBRACE\_R: printf("%d TSMALL\_R %18s\n", linenum, yytext); break;  case TBIGBRACE\_L: printf("%d TLARGE\_L %18s\n", linenum, yytext); break;  case TBIGBRACE\_R: printf("%d TLARGE\_R %18s\n", linenum, yytext); break;  case TMIDBRACE\_L: printf("%d TMID\_L %18s\n", linenum, yytext); break;  case TMIDBRACE\_R: printf("%d TMID\_R %18s\n", linenum, yytext); break;  case TCOLON: printf("%d TCOLON %18s\n", linenum, yytext); break;  case TSEMICOLON: printf("%d TSEMICOLON %18s\n", linenum, yytext); break;  //WHITE SPACE  case TBLANK: printf("%d TBLANK %18s\n", linenum, yytext); break;  case TTAB: printf("%d TTAB %18s\n", linenum, yytext); break;  case TNEWLINE: linenum++; break;  /\*여기부터 따로 정의한 토큰\*/  //ERROR  case TERROR:PrintError(illsp); printf("\tline number[%d] => %s\n", linenum, yytext); break;  case TERROR\_DIGIT\_START: PrintError(illid); printf("\tline number[%d] => %s\n", linenum, yytext); break;  case TERROR\_ILL\_CHAR\_1: PrintError(illsp); printf("\tline number[%d] => %s\n", linenum, yytext); break;  case TERROR\_ILL\_CHAR\_2: PrintError(illsp); printf("\tline number[%d] => %s\n", linenum, yytext); break;  case TLONGERR: PrintError(toolong); printf("\tline number[%d] => %s\n", linenum, yytext); break;  //수  case TREAL: printf("%d TREAL %18s\n", linenum, yytext); break;  case TNUMBER: printf("%d TNUMBER %18s\n", linenum, yytext); break;  case TOCTA:  base = 1;  value = 0;  for (int i = strlen(yytext) - 1; i >= 0; i--) {  value += (yytext[i] - 48) \* base;  base = base \* 8;  }  printf("%d TOCTA %18d\n", linenum, value); break;  case THEXA:  base = 1;  value = 0;  for (int i = strlen(yytext) - 1; i >= 0; i--) {  if (yytext[i] >= '0' && yytext[i] <= '9') {  value += (yytext[i] - 48) \* base;  base = base \* 16;  }  else if (yytext[i] >= 'A' && yytext[i] <= 'F') {  value += (yytext[i] - 55) \* base;  base = base \* 16;  }  }  printf("%d THEXA %18d\n", linenum, value); break;  }  }  PrintHStable();  } |

* 1. PrintError.c

|  |
| --- |
| /\*  PrintError.c- 오류라고 인식된 토큰을 출력할때 어떤 에러메시지를 남길지 관리한다.  programmer-이경연, 박소현  date-19.04.26  \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include "glob.h"  //err에 저장된 값에 따라 다른 에러문이 출력되도록 해주는 함수  void PrintError(ERRORtypes err) {  //String table의 크기보다 더 많은 값들이 저장된 경우  if (err == overst) {  printf("...Error... OVERFLOW");  // PrintHStable();  }  else if (err == toolong) //변수의 길이가 너무 긴 경우  printf("...Error... too long");  //변수 이름이 숫자로 시작되는 경우  else if (err == illid) {  printf("...Error... Start with digit");  }  //정의되지않은 문자가 나올 경우  else if (err == illsp) {  printf("...Error... illegal char");  }  } |

* 1. Hstable.c

|  |
| --- |
| /\*  PrintError.c- 오류라고 인식된 토큰을 출력할때 어떤 에러메시지를 남길지 관리한다.  programmer-이경연, 박소현  date-19.04.26  \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include "glob.h"  //err에 저장된 값에 따라 다른 에러문이 출력되도록 해주는 함수  void PrintError(ERRORtypes err) {  //String table의 크기보다 더 많은 값들이 저장된 경우  if (err == overst) {  printf("...Error... OVERFLOW");  // PrintHStable();  }  else if (err == toolong) //변수의 길이가 너무 긴 경우  printf("...Error... too long");  //변수 이름이 숫자로 시작되는 경우  else if (err == illid) {  printf("...Error... Start with digit");  }  //정의되지않은 문자가 나올 경우  else if (err == illsp) {  printf("...Error... illegal char");  }  } |

* 1. glob.h

|  |
| --- |
| #pragma once  /\*  glob.h- 프로젝트에 필요한 글로벌 변수를 선언  programmer-이경연, 박소현  date-19.04.26  \*/  #define STsize 300 //size of string table  #define HTsize 100 //size of hash table  #define FALSE 0  #define TRUE 1  //letter, digit, 그리고 seperator에 대해 특정 문자가 해당 변수에 속하는지 체크할 때 사용  #define isLetter(ch) (((ch) >= 'a' && (ch) <= 'z') || ((ch) >= 'A' && (ch) <= 'Z') || (ch) == '\_')  #define isDigit(x) (((x)<='9' && (x)>='0'))  #define isSeperator(s) ((s) == ' ' | (s) == '\n' | (s) == '\t' | (s) == ';' | (s) == '.' | (s) == ',' | (s) == '?' | (s) == '!')  typedef struct HTentry \*HTpointer;  typedef struct HTentry {  int index; //index of identifier in ST  HTpointer next; //pointer to next identifier  }HTentry;  //나올 수 있는 여러 종류의 에러타입들을 미리 지정해둔다.  enum errorTypes { noerror, illsp, illid, overst, toolong };  typedef enum errorTypes ERRORtypes;  HTpointer HT[HTsize];  char ST[STsize];  void PrintError(ERRORtypes err);  void PrintHeading();  void PrintHStable();  void ComputeHS(int nid, int nfree);  void LookupHS(int nid, int hscode);  void ADDHT(int hscode);  int nextid ;  int nextfree ;  int hashcode ;  int sameid ;  int found;  ERRORtypes err;  int line\_num;  char \*yytext\_for\_ht;  char input\_char;  int indexyy;  int leng; |

* 1. tokentypes.h

|  |
| --- |
| /\*  tokentypes.h- 토큰들의 종류를 Enum으로 선언한 헤더.  programmer-이경연, 박소현  date-19.04.26  \*/  enum tokentypes {  TEOF,    TCONST, TELSE, TIF, TINT, TRETURN, TVOID, TWHILE,  TPLUS, TMINUS, TMUL, TDIV, TPERCENT,  TADDASSIGN, TSUBASSIGN, TMULASSIGN, TDIVASSIGN, TMODASSIGN, TASSIGN,  TNOT, TAND, TOR,  TEQUAL, TNOTEQU, TLESS, TGREAT, TEQLESS, TEQGREAT,  TINC, TDEC,  TNEWLINE, TBLANK, TTAB,  TBIGBRACE\_L, TBIGBRACE\_R, TMIDBRACE\_L, TMIDBRACE\_R, TSMALLBRACE\_L, TSMALLBRACE\_R,  TCOLON,  TIDENT, TSEMICOLON,TSTRING,TCHAR,    TNUMBER, TREAL, TOCTA, THEXA,  TERROR\_DIGIT\_START, TERROR\_ILL\_CHAR\_1, TERROR\_ILL\_CHAR\_2, TLONGERR, TERROR  }; |

2.No error in the input data file

2.1 Input data\_1

|  |
| --- |
| hi my nickname is ml23is  my height is 175.56 cm  [JANE]  while(true)  //test |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TIDENT hi 0  1 TIDENT my 3  1 TIDENT nickname 6  1 TIDENT is 15  1 TIDENT ml23is 18  2 TIDENT my 25  2 TIDENT height 25  2 TIDENT is 32  2 TREAL 175.56  2 TIDENT cm 32  3 TLARGE\_L [  3 TIDENT JANE 35  3 TLARGE\_R ]  4 TWHILE while  4 TSAMLL\_L (  4 TIDENT true 40  4 TSMALL\_R ) |

2.2 Input data\_2

|  |
| --- |
| The ||bus is red  {Tom} is a (cat)  Sam1! ha%=s a bad //leg  Pam% has;a dog,"test=" |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TIDENT The 0  1 TOR ||  1 TIDENT bus 4  1 TIDENT is 8  1 TIDENT red 11  2 TMID\_L {  2 TIDENT Tom 15  2 TMID\_R }  2 TIDENT is 19  2 TIDENT a 19  2 TSAMLL\_L (  2 TIDENT cat 21  2 TSMALL\_R )  3 TIDENT Sam1 25  3 TNOT !  3 TIDENT ha 30  3 TMODASSIGN %=  3 TIDENT s 33  3 TIDENT a 35  3 TIDENT bad 35  4 TIDENT Pam 39  4 TPERCENT %  4 TIDENT has 43  4 TSEMICOLON ;  4 TIDENT a 47  4 TIDENT dog 47  4 TCOLON ,  4 TSTRING "test=" |

2.3 Input data\_3

|  |
| --- |
| THIS CREature has ( 5 ) legs  ||  dog != {cat || hippo}  return ans  /\*just for fun\*/ |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TIDENT THIS 0  1 TIDENT CREature 5  1 TIDENT has 14  1 TSAMLL\_L (  1 TNUMBER 5  1 TSMALL\_R )  1 TIDENT legs 18  2 TOR ||  3 TIDENT dog 23  3 TNOTEQU !=  3 TMID\_L {  3 TIDENT cat 27  3 TOR ||  3 TIDENT hippo 31  3 TMID\_R }  4 TRETURN return  4 TIDENT ans 37 |

3.With error in the input data file

3.1 Input data\_1

|  |
| --- |
| Let s -= keep th3e exaMples simple !=  0x12 ex\a3Mple && data should == not ha4ve err\_ors  Do you0x23 0x23SUPER it longkdjsljfksdj&\*  '@'@  3Asdsd |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TIDENT Let 0  1 TIDENT s 4  1 TSUBASSIGN -=  1 TIDENT keep 6  1 TIDENT th3e 11  1 TIDENT exaMples 16  1 TIDENT simple 25  1 TNOTEQU !=  2 THEXA 18  2 TIDENT ex 32  ...Error... illegal char line number[2] => \  2 TIDENT a3Mple 35  2 TAND &&  2 TIDENT data 42  2 TIDENT should 47  2 TEQUAL ==  2 TIDENT not 54  2 TIDENT ha4ve 58  2 TIDENT err\_ors 64  3 TIDENT Do 72  3 TIDENT you0x23 75  3 THEXA 35  3 TIDENT SUPER 83  3 TIDENT it 89  ...Error... too long line number[3] => longkdjsljfksdj  ...Error... illegal char line number[3] => &  3 TMUL \*  4 TCHAR '@'  ...Error... illegal char line number[4] => @  ...Error... Start with digit line number[5] => 3Asdsd |

3.2 Input data\_2

|  |
| --- |
| 'string'  's'  "string"  //'STRING'@@ |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  ...Error... illegal char line number[1] => '  1 TIDENT string 0  1 TCHAR '  '  1 TIDENT s 7  ...Error... illegal char line number[1] => '  2 TSTRING "string" |

3.3 Input data\_3

|  |
| --- |
| While(true){  12.55kk  1255kk  "12.55KK"  }  /\*  END TOOOOOLOOOONGGGEKJGKEJGKEJ!!  \*/  longlonglong71616; |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TIDENT While 0  1 TSAMLL\_L (  1 TIDENT true 6  1 TSMALL\_R )  1 TMID\_L {  3 TREAL 12.55  3 TIDENT kk 11  ...Error... Start with digit line number[4] => 1255kk  5 TSTRING "12.55KK"  6 TMID\_R }  ...Error... too long line number[8] => longlonglong71616  8 TSEMICOLON ; |

4.Data file given from professor

4.1 Input data\_1

|  |
| --- |
| int main(void) {  //2019 compiler!!    const int a = 8;  int A = 0x12 + 011; //hex + oct  while(A >= 10){  }  return 0;  } |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TINT int  1 TIDENT main 0  1 TSAMLL\_L (  1 TVOID void  1 TSMALL\_R )  1 TMID\_L {  4 TCONST const  4 TINT int  4 TIDENT a 5  4 TASSIGN =  4 TNUMBER 8  4 TSEMICOLON ;  5 TINT int  5 TIDENT A 7  5 TASSIGN =  5 THEXA 18  5 TPLUS +  5 TOCTA 9  5 TSEMICOLON ;  6 TWHILE while  6 TSAMLL\_L (  6 TIDENT A 9  6 TEQGREAT >=  6 TNUMBER 10  6 TSMALL\_R )  6 TMID\_L {  7 TMID\_R }  8 TRETURN return  8 TNUMBER 0  8 TSEMICOLON ;  9 TMID\_R } |

4.2 Input data\_2

|  |
| --- |
| #include <stdio.h>  void main(VOID) {  //string str1 = "";  string str1 = "\"";  string str2 = "blah blah";  if( !flag )  printf("%s\n", str1 + str2 + str1;);  } |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TIDENT While 0  1 TSAMLL\_L (  1 TIDENT true 6  1 TSMALL\_R )  1 TMID\_L {  3 TREAL 12.55  3 TIDENT kk 11  ...Error... Start with digit line number[4] => 1255kk  5 TSTRING "12.55KK"  6 TMID\_R }  ...Error... too long line number[8] => longlonglong71616  8 TSEMICOLON ; |

4.3 Input data\_3

|  |
| --- |
| /\* average of 2integers \*/  float average(int a, int b)  {  int tooLongIdent = 0X13;  float result = (a + b) / 2.0000f;  return result;  } |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  2 TIDENT float 0  2 TIDENT average 6  2 TSAMLL\_L (  2 TINT int  2 TIDENT a 14  2 TCOLON ,  2 TINT int  2 TIDENT b 16  2 TSMALL\_R )  3 TMID\_L {  4 TINT int  ...Error... too long line number[4] => tooLongIdent  4 TASSIGN =  4 THEXA 19  4 TSEMICOLON ;  5 TIDENT float 18  5 TIDENT result 18  5 TASSIGN =  5 TSAMLL\_L (  5 TIDENT a 25  5 TPLUS +  5 TIDENT b 25  5 TSMALL\_R )  5 TDIV /  5 TREAL 2.0000  5 TIDENT f 25  5 TSEMICOLON ;  6 TRETURN return  6 TIDENT result 27  6 TSEMICOLON ;  7 TMID\_R } |

4.4 Input data\_4

|  |
| --- |
| 2 x 12=24 >011  31.2456~~~IsBiggerThan 30!  , Visit"\\\\\\." + "ewha" + "." + "ac.kr" ^^  else~  "eportal //.ewha.ac.kr |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TNUMBER 2  1 TIDENT x 0  1 TNUMBER 12  1 TASSIGN =  1 TNUMBER 24  1 TGREAT >  1 TOCTA 9  2 TREAL 31.2456  ...Error... illegal char line number[2] => ~  ...Error... illegal char line number[2] => ~  ...Error... illegal char line number[2] => ~  ...Error... too long line number[2] => IsBiggerThan  2 TNUMBER 30  2 TNOT !  3 TCOLON ,  3 TIDENT Visit 2  3 TSTRING "\\\\\\."  3 TPLUS +  3 TSTRING "ewha"  3 TPLUS +  3 TSTRING "."  3 TPLUS +  3 TSTRING "ac.kr"  ...Error... illegal char line number[3] => ^  ...Error... illegal char line number[3] => ^  4 TELSE else  ...Error... illegal char line number[4] => ~  ...Error... illegal char line number[5] => "  5 TIDENT eportal 8 |

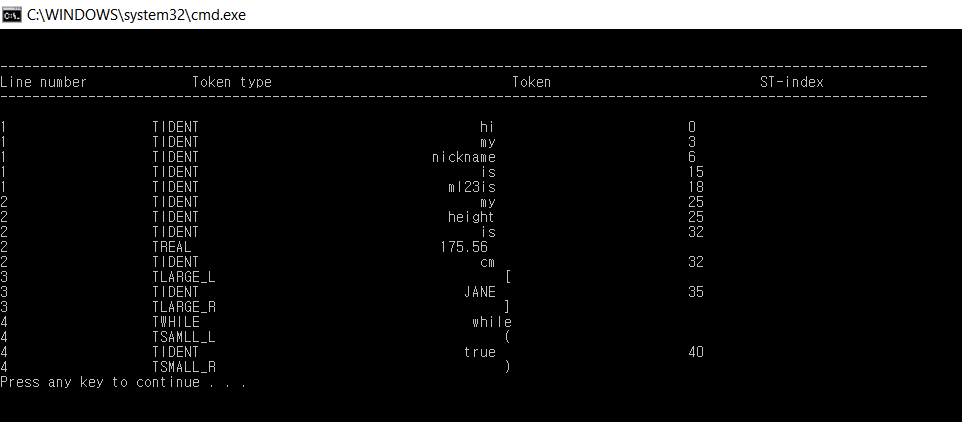
4.5 Input data\_

|  |
| --- |
| int 2arr[2] = {0xAB, 0Xc2};  void main(){    SU@M( arr[0], arr[1]); // add  "} 21. |
| --------------------------------------------------------------------------------------------------------------------  Line number Token type Token ST-index  --------------------------------------------------------------------------------------------------------------------  1 TINT int  ...Error... Start with digit line number[1] => 2arr  1 TLARGE\_L [  1 TNUMBER 2  1 TLARGE\_R ]  1 TASSIGN =  1 TMID\_L {  1 THEXA 171  1 TCOLON ,  1 THEXA 2  1 TMID\_R }  1 TSEMICOLON ;  2 TVOID void  2 TIDENT main 0  2 TSAMLL\_L (  2 TSMALL\_R )  2 TMID\_L {  4 TIDENT SU 5  ...Error... illegal char line number[4] => @  4 TIDENT M 8  4 TSAMLL\_L (  4 TIDENT arr 10  4 TLARGE\_L [  4 TNUMBER 0  4 TLARGE\_R ]  4 TCOLON ,  4 TIDENT arr 14  4 TLARGE\_L [  4 TNUMBER 1  4 TLARGE\_R ]  4 TSMALL\_R )  4 TSEMICOLON ;  ...Error... illegal char line number[5] => "  5 TMID\_R }  5 TNUMBER 21  ...Error... illegal char line number[5] => . |

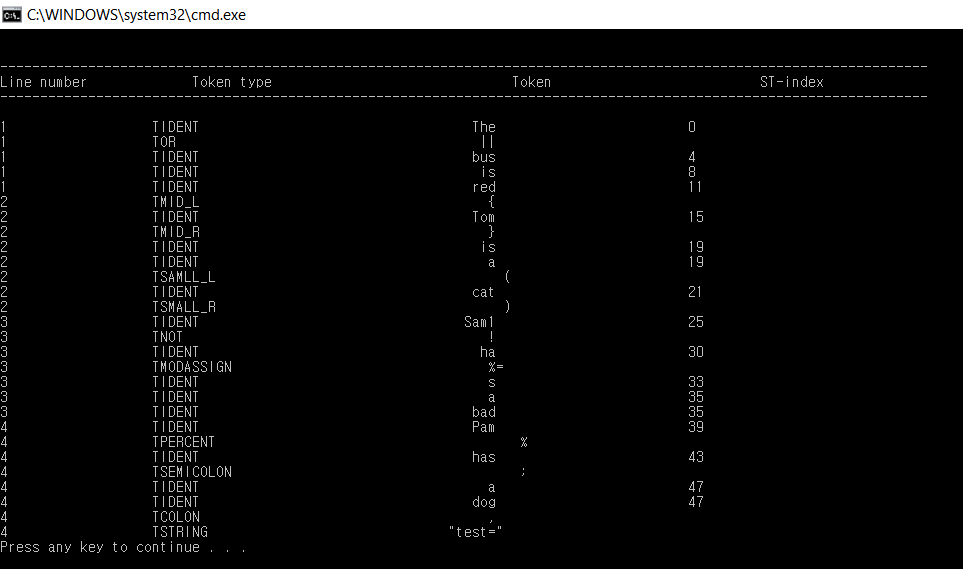
5. Screenshots

1. No error in the input data file

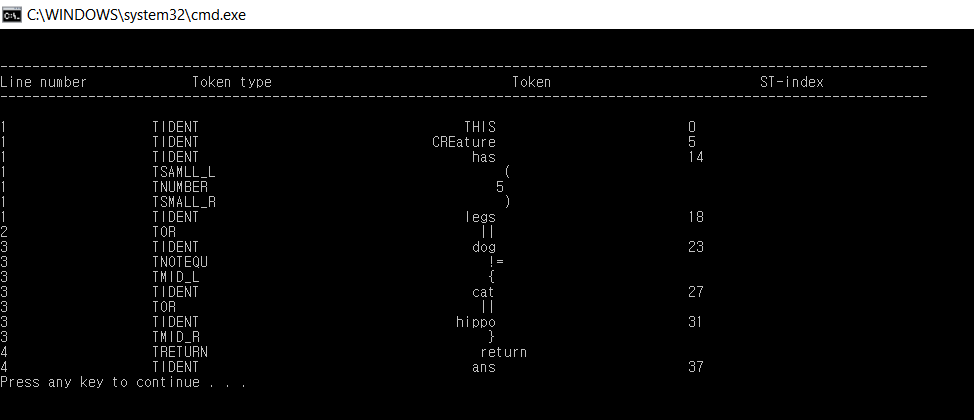
1.1 Input\_data1



1.2 Input\_data2

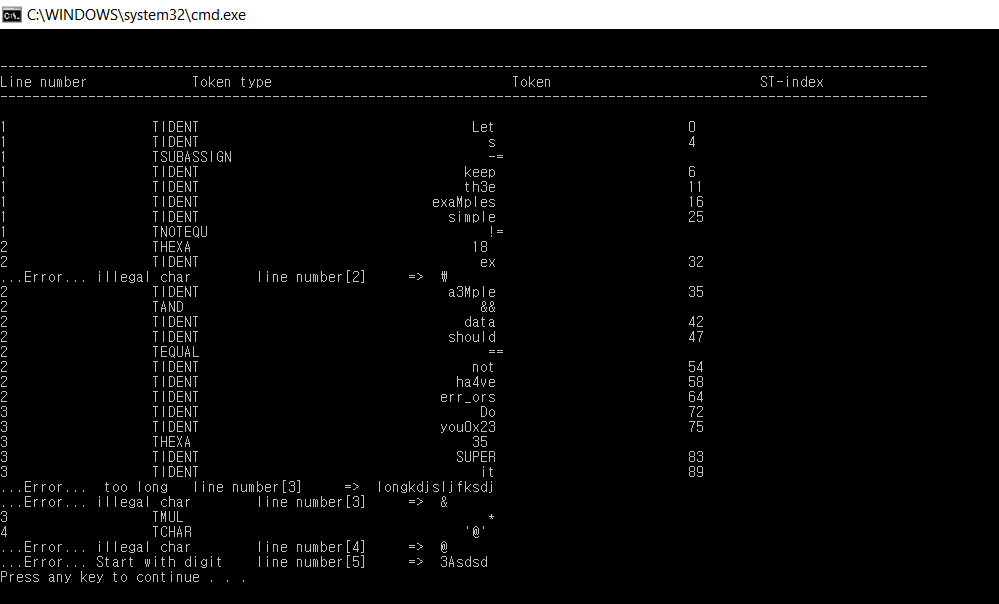


1.3 Input\_data3



2. With error in the input data file

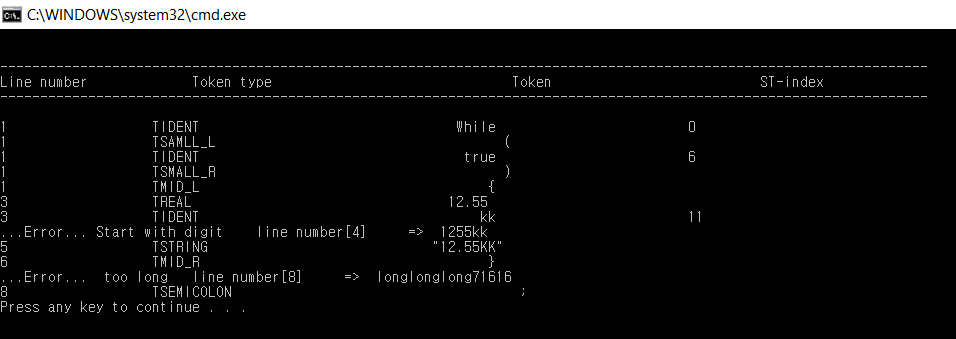
2.1 Input\_data1



2.2 Input\_data2

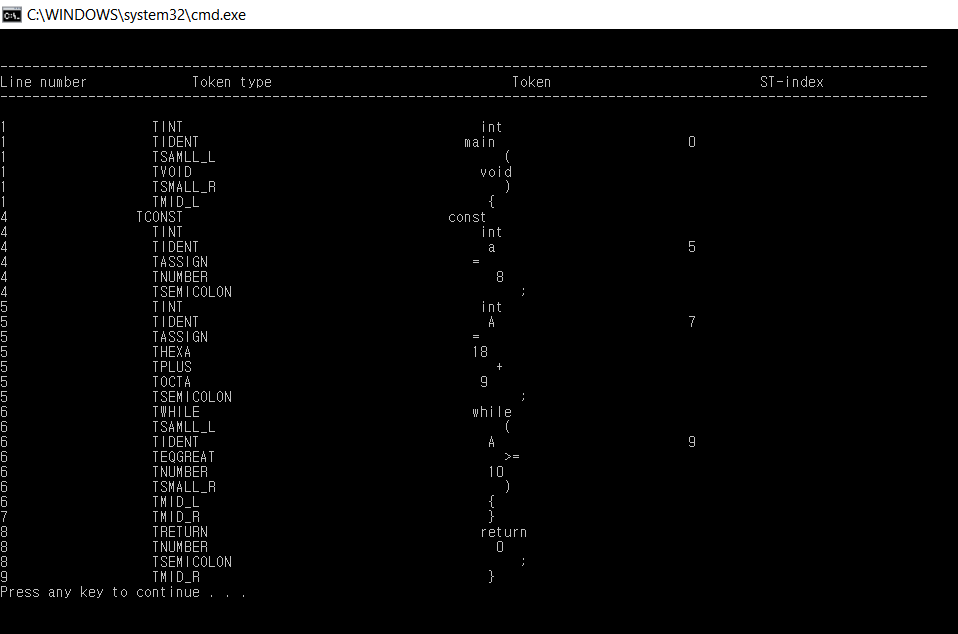


2.3 Input\_data3

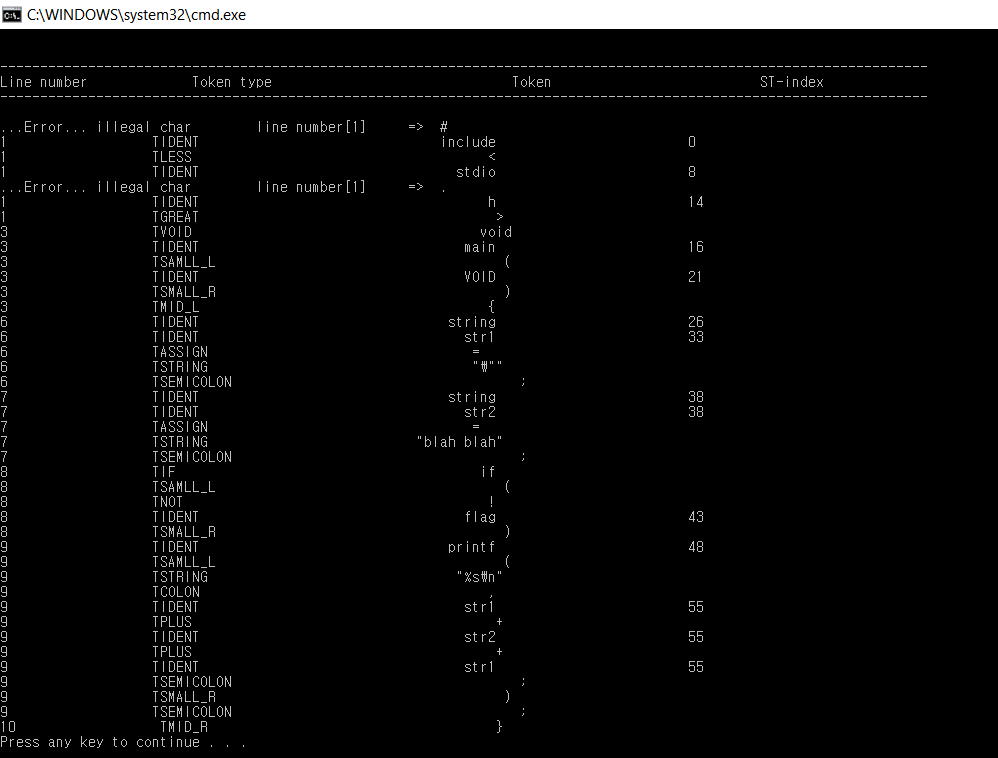


3. Data file given from Professor

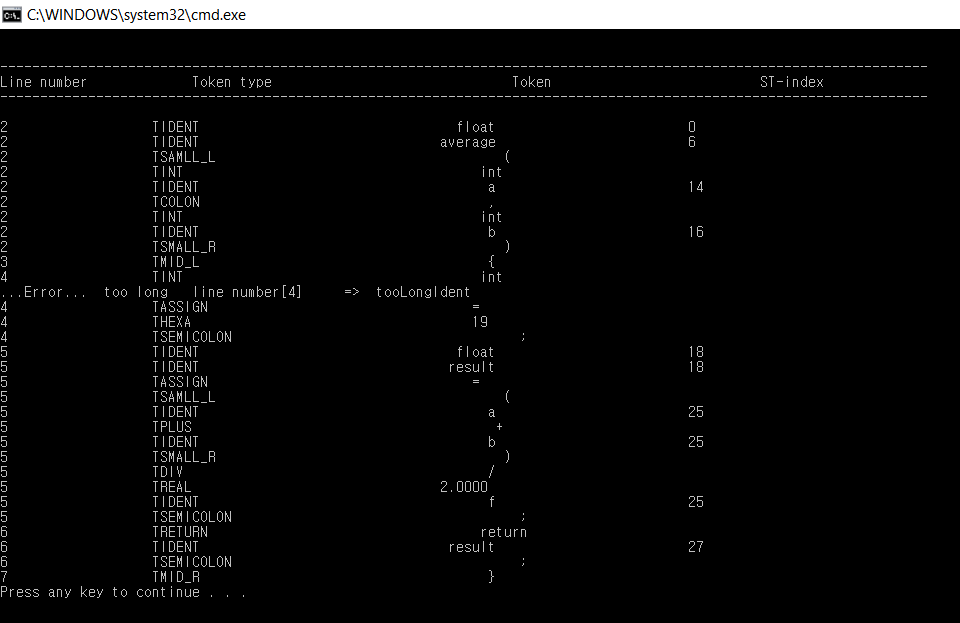
3.1 testdata1



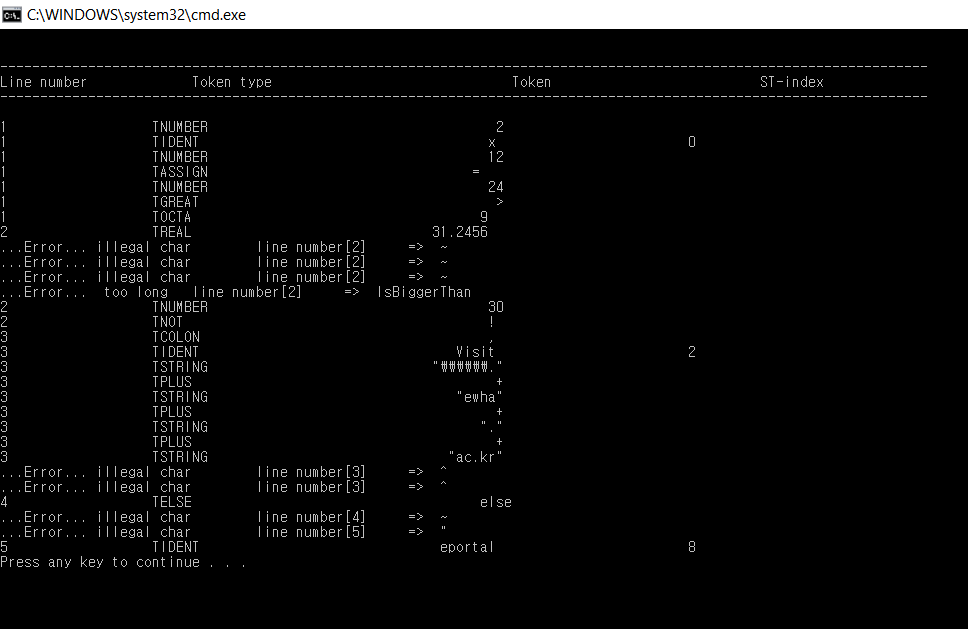
3.2 testdata2



3.3 testdata3



3.4 testdata4



3.5 testdata5

