%{

  #include<stdio.h>

  #include<string.h>

  char lexema[255];

  void yyerror(char \*);

%}

// Especificamos los tokens

%token CORREO

// Especificamos la gramatica

%%

instruccion: CORREO;

instruccion: ;

%%

void yyerror(char \*msg) {

  printf("error: %s", msg);

}

// Especificamos las reglas de los tokens

int yylex() {

  char c;

  while(1) {

    c = getchar();

    if(c == '\n') continue;

    if(c == ' ') continue;

    // Token CORREO

    if(isalpha(c)) {

      int i = 0;

      do {

        lexema[i++] = c;

        c = getchar();

      } while(isalpha(c));

      char dominio[] = "@uni.edu.pe";

      int j = 0;

      if(c == dominio[j]) {

        do {

          lexema[i++] = c;

          c = getchar();

          j++;

        } while(c == dominio[j] && j < 11);

        if (j == 11) {

          ungetc(c, stdin);

          lexema[i] == 0;

          return CORREO;

        }

      }

    }

    return c;

  }

}

int main() {

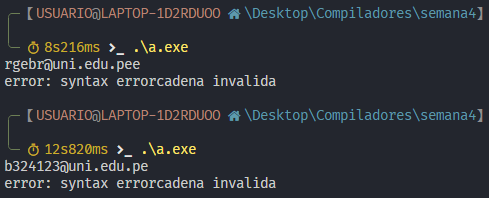
  if(!yyparse()) printf("cadena valida\n");

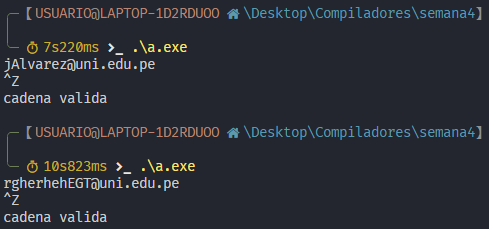
  else printf("cadena invalida\n");

  return 0;

}

Código valido solo para token CORREO (letras seguido de @uni.edu.pe)





%{

  #include<stdio.h>

  #include<string.h>

  char lexema[255];

  void yyerror(char \*);

%}

// Especificamos los tokens

%token NUMERO

// Especificamos la gramatica

%%

instruccion: NUMERO;

instruccion: ;

%%

void yyerror(char \*msg) {

  printf("error: %s", msg);

}

// Especificamos las reglas de los tokens

int yylex() {

  char c;

  while(1) {

    c = getchar();

    if(c == '\n') continue;

    if(c == ' ') continue;

    // Token NUMERO

    if(isdigit(c) && c != '0') {

      int i = 0;

      lexema[i++] = c;

      if (c == '1') {

        c = getchar();

        if(c == '0') {

          lexema[i++] = c;

          c = getchar();

          if(c == '.') {

            lexema[i++] = c;

            c = getchar();

            // Despues del . continuan varios 0s

            if(c == '0') {

              do {

                lexema[i++] = c;

                c = getchar();

              } while(c == '0');

              // Despues una E

              if(c == 'E') {

                // Despues varios digitos

                do {

                  lexema[i++] = c;

                  c = getchar();

                } while(isdigit(c));

                ungetc(c, stdin);

                lexema[i] == 0;

                return NUMERO;

              }

            }

          }

        }

        if(c == '.') {

          lexema[i++] = c;

          c = getchar();

          // Despues del . continuan varios digitos

          if(isdigit(c)) {

            do {

              lexema[i++] = c;

              c = getchar();

            } while(isdigit(c));

            // Despues una E

            if(c == 'E') {

              // Despues varios digitos

              do {

                lexema[i++] = c;

                c = getchar();

              } while(isdigit(c));

              ungetc(c, stdin);

              lexema[i] == 0;

              return NUMERO;

            }

          }

        }

      } else {

        // Continua un . para que numero real

        // este entre 1 y 10

        if(c == '.') {

          lexema[i++] = c;

          c = getchar();

          // Despues del . continuan varios digitos

          if(isdigit(c)) {

            do {

              lexema[i++] = c;

              c = getchar();

            } while(isdigit(c));

            // Despues una E

            if(c == 'E') {

              // Despues varios digitos

              do {

                lexema[i++] = c;

                c = getchar();

              } while(isdigit(c));

              ungetc(c, stdin);

              lexema[i] == 0;

              return NUMERO;

            }

          }

        }

      }

    }

    return c;

  }

}

int main() {

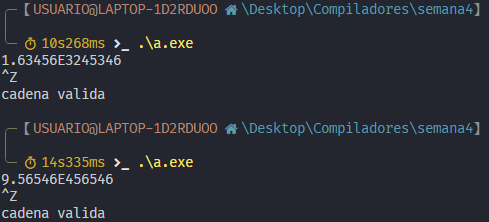
  if(!yyparse()) printf("cadena valida\n");

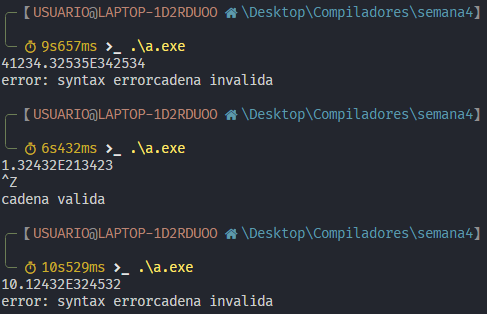
  else printf("cadena invalida\n");

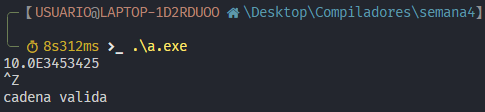
  return 0;

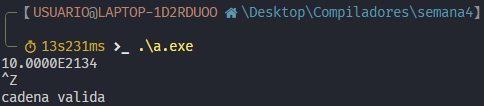
}

Código valido solo para token NUMERO (Real entre 1 y 10 seguido de E seguido de numero)









Código valido para ambos tokens

%{

  #include<stdio.h>

  #include<string.h>

  char lexema[255];

  void yyerror(char \*);

%}

// Especificamos los tokens

%token CORREO NUMERO

// Especificamos la gramatica

%%

instruccion: expr instruccion;

instruccion: ;

expr: CORREO | NUMERO;

%%

void yyerror(char \*msg) {

  printf("error: %s", msg);

}

// Especificamos las reglas de los tokens

int yylex() {

  char c;

  while(1) {

    c = getchar();

    if(c == '\n') continue;

    if(c == ' ') continue;

    // Token CORREO

    if(isalpha(c)) {

      int i = 0;

      do {

        lexema[i++] = c;

        c = getchar();

      } while(isalpha(c));

      char dominio[] = "@uni.edu.pe";

      int j = 0;

      if(c == dominio[j]) {

        do {

          lexema[i++] = c;

          c = getchar();

          j++;

        } while(c == dominio[j] && j < 11);

        if (j == 11) {

          ungetc(c, stdin);

          lexema[i] == 0;

          return CORREO;

        }

      }

    }

    // Token NUMERO

    if(isdigit(c) && c != '0') {

      int i = 0;

      lexema[i++] = c;

      if (c == '1') {

        c = getchar();

        if(c == '0') {

          lexema[i++] = c;

          c = getchar();

          if(c == '.') {

            lexema[i++] = c;

            c = getchar();

            // Despues del . continuan varios 0s

            if(c == '0') {

              do {

                lexema[i++] = c;

                c = getchar();

              } while(c == '0');

              // Despues una E

              if(c == 'E') {

                // Despues varios digitos

                do {

                  lexema[i++] = c;

                  c = getchar();

                } while(isdigit(c));

                ungetc(c, stdin);

                lexema[i] == 0;

                return NUMERO;

              }

            }

          }

        }

        if(c == '.') {

          lexema[i++] = c;

          c = getchar();

          // Despues del . continuan varios digitos

          if(isdigit(c)) {

            do {

              lexema[i++] = c;

              c = getchar();

            } while(isdigit(c));

            // Despues una E

            if(c == 'E') {

              // Despues varios digitos

              do {

                lexema[i++] = c;

                c = getchar();

              } while(isdigit(c));

              ungetc(c, stdin);

              lexema[i] == 0;

              return NUMERO;

            }

          }

        }

      } else {

        // Continua un . para que numero real

        // este entre 1 y 10

        if(c == '.') {

          lexema[i++] = c;

          c = getchar();

          // Despues del . continuan varios digitos

          if(isdigit(c)) {

            do {

              lexema[i++] = c;

              c = getchar();

            } while(isdigit(c));

            // Despues una E

            if(c == 'E') {

              // Despues varios digitos

              do {

                lexema[i++] = c;

                c = getchar();

              } while(isdigit(c));

              ungetc(c, stdin);

              lexema[i] == 0;

              return NUMERO;

            }

          }

        }

      }

    }

    return c;

  }

}

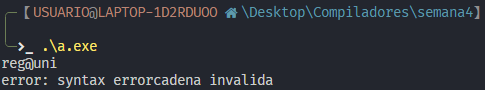
int main() {

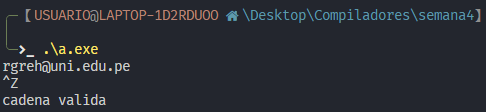
  if(!yyparse()) printf("cadena valida\n");

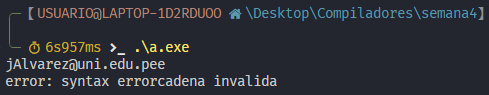
  else printf("cadena invalida\n");

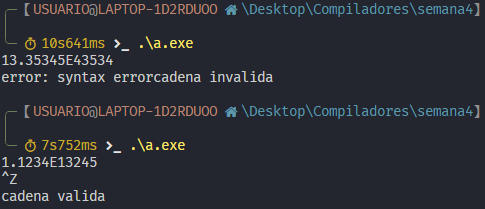
  return 0;

}

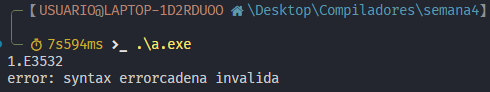


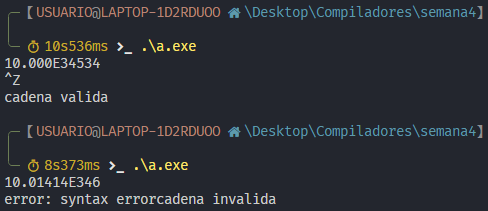






Considerando que después del punto debe continuar al menos un numero





Penadillo Lazares Wenses Johan