Instituto Tecnológico de Costa Rica

Compiladores e Intérpretes

Proyecto2: Análisis Sintáctico

Prof: Francisco Torres Rojas

Jason Barrantes A. 2015048456
Randy Morales G. 2015085446

II Semestre 2017

CONTENIDOS

BISON Y FASE DE PARSING

BISON

BISON es un proyecto creado por GNU, por lo que es la versión libre de YACC. Es un programa que genera analizadores sintácticos con propósitos generales. Este está disponible para cualquier sistema operativo y se usa acompañado de **Flex**. Convierte la descripción de un lenguaje, escrita con una gramática libre de contexto (CFG), en un programa en C, C++ o Java que realiza el análisis sintáctico. Para la utilización de Bison, es necesaria tener la gramática a analizar.

Al compilarse el archivo Bison se generan dos archivos , un **archivo .h** y otro .c. El archivo .h debe incluirse en el archivo de Lex/Flex.

ELEMENTOS

El archivo de codificación de Bison tiene cuatro secciones:

%

Declaraciones C

%

Declaraciones Yacc/Bison

%Declaracion de Token

%%

Reglas de la gramática

%%

Código C

ELEMENTOS

%{, %} sirven para delimitar el encabezado, donde usualmente se colocan bibliotecas

%% sirven para indicar cuales son las reglas

Y por último la programación en C o llamadas a funciones en caso de ser necesario.

%Token Se utiliza para definir los símbolos no terminales (tokens) de la gramática.

%token NOMBRE_TOKEN

Por convenio, los nombres de los tokens se escriben en mayúsculas y se pueden agrupar varios tokens en una línea si tienen en mismo tipo.

CREADORES

Bison fue escrito en un principio por Robert Corbett; Richard Stallman lo hizo compatible con Yacc y Wilfred Hansen de la Carnegie Mellon University añadió soporte para literales multicaracter y otras características.

Richard Stallman

Parsing

El parsing es la segunda fase de un compilador. La fase de parsing es el proceso donde se toma el conjunto de tokens producidos por la fase de análisis léxico y se genera un **árbol de sintaxis**. Luego este se revisa de acuerdo a la gramática formal del lenguaje definido y se verifica que las expresiones construidas por el arreglo de tokens sean sintacticamente correctos.

Código fuente

```
1
2
3
4
5
6       static signed char a[] = {0x69,
7     110, 118, 97, 108, 105,  0x64, 1-1,
8   0x6d, 111, 118,  101, 1<<1<<1<<1<<1<<1,
9   114,  105, 0x6e,  103,  32, 'o'/3, 100,
10  32, 102, 114, 111,  0x6d, 32, 115, 116,
11  97, 100-001, 107, 32,37, 2*'2', '@'>>1,
12  116, ' + ' + ','w'-'W',115, 0x74,
13  97, 3*'!', 107, 'q' - 'Q', 37, 10*'\n',
14  10, 0}, * b = a + (1<<1<<1<<1), * w, x,
15  *q, c, r; int main(int d, char *e []) {
16  return q = (signed char *)(e+1+1), (r =
17  e[0] && e[1] ? 0 : 0 * puts (a) + 1) ||
18  (r = e[1<<1] && d != 1 <<1 && 0 * puts(
19  a) + 1) || e[1- -1] ||  (r = atoi(e[1])
20  < -0200 || atoi (e [1]) > 0x7f || ( x =
21  atoi( e[1] ) ) == 0 ? 0 * puts(a) + 1 :
22  0) || e [1- -1] || (x- -x > 1-1 ? (q[0]
```

```
23                 = x, q[1] = q[3] = 1, q[2] = 2) : (
24                   memset ( w = ( signed  char * )
25
26
27
28
29
30
31
32                     malloc(-x), '1', -x), puts (w),
33                   q[0] = x, q[1] = '0', q[2] = q[3] =
34                 0)), r || (q[3] ? (c = 6 - q[1] - q[2],
35                 (q[0] != 1) ? q[0]-- , d = q[2], q[2] =
36                 c, main(2, e), c = q[2], q[2] = d, q[0]
37                 ++ : 0, printf(b, q[0], q[1], q[2]), (q
38                 [0] != 1) ? q[0]--, d = q[1],q[1] = c ,
39                 main(2, e), c = q[1], q[1] = d, q[0] ++
40                 : 0) : - 1 - q[0] - 1 == 0 ? (w[- x - 1
41                 - (q[1] & 1 ^ 1)] = q[1], puts (w), w [
42                 - x - 1 - (q[1] & 1)] = q[1], puts(w) )
43                 : - 1 - q[0] == 0 ?  (w[- x - 1] = q[ 1
44                   ], puts(w)) : (q[0] += 1 + ( q[1] & 1
```

```
                        ^ 1), main(2, e), q[0] -= 1 + ( q
                              [1] & 1 ^ 1), q[1] & 1 ? (q
                                 [0]+=1+1,  q[1]^=1, main
                                 (2, e), q[1]^=1, q[0]-=1
                                 +1) : 0, w[q[0] - x] = q
                                [1], puts(w), q[1] & 1 ?
                               0 : (q[0]+=1+1, q[1]^=1,
                            main (2, e), q[1]^=1, q
                          [0]-=1+1), q[0] += 1 +
                      (q[1] & 1),main(2,e)
                   , q[0] -= 1 + (q[1]
                 & 1) ) ), r; }


                  float clauses(){
  float x = (0.5 +
    1.23) - 2.145 * (7451 / 124.6)
                      + 987 * 41 - 7123;
  float y = (0.5 + 1.23) - 2.145 *
                 (41 * 7 + (41256 * 4551)) + 9 * 11245 - 23;
  float z = (x * y) / (y * y) + 12000;
```

```
67    float R = (0.5 + 1.23) - 2.145 *
68
69
70                    (7451 / 124.6) + 987 * 41
71                    - 7123;
72
73    float P = (0.5 + 1.23) - 2.145 * (41 * 7
74              + (41256 * 4551)) + 9 * 11245 - 23;
75    float G = (x * y) / (y * y) + 12000;
76    float S3 = (0.5 + 1.23) - 2.145 *
77
78
79    (7451 / 124.6) + 987 * 41 - 7123;
80
81    float A0 = (0.5 + 1.23) - 2.145 * (41 * 7 +
82                                       (41256 * 4551)) + 9 *
      11245 - 23;
83    float P1 = (x * y) / (y * y) + 12000;
84
85    float R0 = (0.5 + 1.23) -
86    2.145 * (7451 / 124.6) + 987 * 41 - 7123;
87
```

```
88    float I = (0.5 + 1.23) - 2.145 * (41 * 7 + (41256 * 4551))
89
90
91                                                  + 9 * 11245 -
        23;
92    float S = (x * y) / (y * y) + 12000;
93    float S1 = (0.5 + 1.23) - 2.145 *
94
95
96    (7451 / 124.6) + 987 * 41 - 7123;
97    float A = (0.5 + 1.23) - 2.145 * (41 *
98
99
100     7 + (41256 * 4551)) + 9 * 11245 - 23;
101   float z1 = (x * y) / (y * y) + 12000;
102   float x2 = (0.5 + 1.23) - 2.145 *
103
104
105   (7451 / 124.6) + 987 * 41 - 7123;
106   float y3 = (0.5 + 1.23) - 2.145 *
107
108                       (41 * 7 + (41256 * 4551)) + 9 * 11245
```

```
    - 23;
109
110
111                     float z4 = (x * y) / (y * y) + 12000;
112
113   return x + y + z + R + P + G + S3 + A0 + P1;
114 }
115
116
117 void include(){
118 char *step0 = "pixa"; char *makeF0 = "Adapted to the files";
119
120
121
122   char *splint0 = "";
123 char *step1 = "pixa";
124
125
126           char *makeF1 = "Adapted to the files";
127           char *splint1 = "know";
128 char *step2 = "pixa"; char *makeF2 = "Adapted to the files";
129
```

```
130
131 char *splint2 = "know";
132 char *step3 = "pixa";
133
134
135 char *makeF3 = "Adapted to the files";
136
137
138       char *splint3 = "know";
139 char *step4 = "pixa"; char *makeF4 = "Adapted to the files";
140
141
142 char *splint4 = "know";
143 }
144
145 static int check_type(){
146
147           char *step0 = "pixa";
148
149     char *makeF0 = "Adapted to the files";  char *splint0 =
      "know";
150 char *step1 = "pixa";
```

```
151
152             char *makeF1 = "Adapted to the files";
153
154
155
156                         char *splint1 = "know";
157 char *step2 = "pixa";
158
159 char *makeF2 = "Adapted to the files";
160
161         char *splint2 = "know";
162 char *step3 = "pixa"; char *makeF3 = "Adapted to the files";
          char *splint3 = "know";
163 char *step4 = "pixa"; char *makeF4 = "Adapted to the files";
          char *splint4 = "know";
164 }
165
166 void count(){
167 int count0 = 0; int count1 = 0; int count2 = 0; int count3 =
        0;
168
169
```

```
170             int count4 = 0; int count5 = 0; int count6 = 0;
171 int count10 = 0;
172
173             int count11 = 0; int count22 = 0;
174
175
176 int count320 = 0; int count415 = 0; int count51 = 0; int
        count61 = 0;
177 int count20 = 0; int count21 = 0;
178
179
180 int count325 = 0; int count330 = 0; int count421 = 0; int
        count52 = 0; int count62 = 0;
181 int count30 = 0; int count311 = 0;
182
183
184 int count32 = 0; int count340 = 0; int count33 = 0; int
        count53 = 0; int count63 = 0;
185 int count40 = 0; int count411 = 0;
186
187
188
```

```
189 int count525 = 0; int count350 = 0; int count44 = 0; int
        count54 = 0; int count64 = 0;
190 int count50 = 0; int count512 = 0; int count626 = 0; int
        count360 = 0; int count45 = 0; int count55 = 0; int
        count65 = 0;
191 char *step0 = "pixa";
192
193
194         char *makeF0 = "Adapted to the files";  char *splint0
        = "";
195 char *step1 = "pixa"; char *makeF1 = "Adapted to the files";
196
197
198           char *splint1 = "know";
199 char *step2 = "pixa"; char *makeF2 = "Adapted to the files";
200
201
202
203         char *splint2 = "know";
204 char *step3 = "pixa"; char *makeF3 = "Adapted to the files";
          char *splint3 = "know";
205 char *step4 = "pixa"; char *makeF4 = "Adapted to the files";
```

```
        char *splint4 = "know";
206 }
207
208 void printLexicalErrors(){
209 char *step0 = "pixa"; char *makeF0 = "Adapted to the files";
210
211
212
213                                 char *splint0 = "";
214 char *step1 = "pixa"; char *makeF1 = "Adapted to the files";
215
216
217             char *splint1 = "know";
218 char *step2 = "pixa"; char *makeF2 = "Adapted to the files";
        char *splint2 = "know";
219 char *step3 = "pixa"; char *makeF3 = "Adapted to the files";
220
221             char *splint3 = "know";
222 char *step4 = "pixa"; char *makeF4 = "Adapted to the files";
        char *splint4 = "know";
223
224 }
```

```
225
226 int yywrap(){
227 int count0 = 0; int count1 = 0;
228
229
230
231                 int count2 = 0; int count3 = 0; int count4 = 0
      ;
232
233
234
235
236
237
238
239                 int count5 = 0; int count6 = 0;
240 int count10 = 0;
241
242
243
244 int count11 = 0; int
245
```

```
246
247         count22 = 0; int count320 = 0;
248
249
250
251
252             int count415 = 0; int count51 = 0; int count61 = 0
    ;
253 int count20 = 0; int count21 = 0;
254
255
256 int count325 = 0; int count330 = 0; int count421 = 0; int
    count52 = 0; int count62 = 0;
257 int count30 = 0; int count311 = 0;
258
259
260 int count32 = 0; int count340 = 0; int count33 = 0; int
    count53 = 0; int count63 = 0;
261 int count40 = 0; int count411 = 0; int count525 = 0;
262
263
264             int count350 = 0; int count44 = 0; int count54 = 0
```

```
    ;
                        int count64 = 0;
int count50 = 0; int count512 = 0;


  int count626 = 0; int count360 = 0; int count45 = 0; int
    count55 = 0; int count65 = 0;
}

void printErrors(){
char *step2 = "pizza";
}

void write_infile

(char string)

        {
  int size =
```

```
285             100;
286   char messageStore[100];
287   for(int i=0;
288
289
290     i < size; i++){
291
292                     messageStore[i]
293
294
295
296     = string;
297   }
298 }
299
300 void luthor(){
301   int r=0,x,y=0,n          =              0     ;
302    char        v   [
303    100     ]                      ,  s;
304 int t[100      ]                       ,
305  u[100                ]     ;  u
306      [                   0
```

```
]=ftell(stdin          )        ;
while((v     )                    )
    {t  [          r ]=
strlen                    (
  v);y=
    t[          r                ]
    >y              ?          t [
      r           ]     :    y;
 u[++r    ]      =
ftell(stdin )    ;                              }
 while            (                      n
     <>                 y          )
{for(x=0                ;          x
<r;x++) {                              s
  =' '         ;
 if(n<                t         [
     x               ]      )     {
fseek(stdin     ,u        [      x     ]
    +n    ,  0 )                        ;
scanf("%c",                          &
                              s         )
      ;                               }
```

```
329 printf("%c",    s                          )
330                                    ;       }
331 printf("\n");                      n
332     ++    ;                }
333 }
334
335 void initValues(){
336   int ax0,bx0,cx0,dx0,ex0,fx0,gx0,
337   hx0,ix0,jx0,kx0,lx0,mx0,nx0,ox0,
338   px0,qx0,rx0,sx0,tx0,ux0,vx0,wx0,
339   xx0,yx0,zx0;
340   int ax1,bx1,cx1,dx1,ex1,fx1,gx1,
341   hx1,ix1,jx1,kx1,lx1,mx1,nx1,ox1,px1,
342                qx1,rx1,sx1,tx1,ux1,vx1,wx1,xx1,yx1,zx1;
343   int ax2,bx2,cx2,dx2,ex2,fx2,gx2,
344   hx2,ix2,jx2,                      kx2,lx2,mx2,nx2,ox2,px2,
345   qx2,rx2,sx2,tx2,ux2,vx2,wx2,xx2,yx2,zx2;
346   int ax3,bx3,cx3,dx3,ex3,fx3,gx3,
347   hx3,ix3,jx3,kx3,lx3,mx3,nx3,ox3,px3,qx3,
348   rx3,sx3,tx3,ux3,vx3,              wx3,xx3,yx3,zx3;
349   int ax4,bx4,cx4,dx4,ex4,fx4,gx4,hx4,ix4,jx4,kx4,lx4,mx4,
350     nx4,ox4,px4,qx4,rx4,sx4,tx4,ux4,vx4,wx4,xx4,yx4,zx4;
```

```
351     int ax5,bx5,                    cx5,dx5,ex5,fx5,gx5,hx5,ix5,
          jx5,kx5,lx5,mx5,
352     nx5,ox5,px5,qx5,rx5,sx5,tx5,ux5,vx5,wx5,xx5,yx5,zx5;
353     int ax6,bx6,cx6,dx6,            ex6,fx6,gx6,hx6,ix6,jx6,kx
          6,lx6,mx6,
354     nx6,ox6,px6,qx6,rx6,sx6,tx6,ux6,vx6,wx6,xx6,yx6,zx6;
355     int ax7,bx7,cx7,dx7,ex7,        fx7,gx7,hx7,ix7,jx7,kx7,lx
          7,mx7,
356     nx7,ox7,px7,            qx7,rx7,sx7,tx7,ux7,vx7,wx7,xx7,yx
          7,zx7;
357     int ax8,bx8,cx8,dx8,ex8,fx8,gx8,hx8,ix8,jx8,kx8,lx8,mx8,
358     nx8,ox8,px8,qx8,rx8,sx8,        tx8,ux8,vx8,wx8,xx8,yx8,zx
          8;
359     int ax9,bx9,cx9,dx9,ex9,        fx9,gx9,hx9,ix9,jx9,kx9,lx
          9,mx9,
360     nx9,ox9,px9,qx9,rx9,sx9,tx9,ux9,vx9,wx9,xx9,yx9,zx9;
361
362 char a19,b19,c19,
363     d19,e19,,f19,g19,h19,i19,j19,
364
365
366
```

```
367     k19,l19,m19,n19,o19,p19
368     ,q19,r19,s19,t19,u19,v19,w19,x19,y19,z19;
369
370   char  af0,bf0,cf0,df0,,ef0,ff0,gf0,hf0,if0,
371
372
373     jf01,kf0,lf0,mf0,nf0,of0;
374
375     char  a1,b1,c1,d1,e1,f1,g1,h1,i1,
376
377
378     j11,k1,l1,m1,n1,o1,
379
380
381
382       p1,q1,r1,s1,t1,u1,v1,w1,x1,y11,z1;
383     char  a2,b2,c2,d2,e2,f2,
384
385
386
387
388                 g2,h2,i2,j2,k2,l2,m2,n2,o2,p2,q2,r2,s2,t2,u2,v2,
```

```
    w2,x2,y2,z2;
389 char a3,b3,c3,d3,e3,f3,g3,h3,i3,j3,
390 k3,l3,m3,n3,o3,p3,q3,r3,s3,t3,u3,v3,w3,x3,y3,z3;
391 char a4,b4,c4,d4,e4,f4,g4,h4,i4,j4,k4,
392
393
394 l4,m4,n4,o4,p4,q4,
395
396
397
398
399                     r4,s4,t4,u4,v4,w4,x4,y4,z4;
400 char a5,b5,c5,d5,e5,                        f5,g5,h5,i5,j5,k
    5,l5,
401
402 m5,n5,o5,p5,q5,
403
404
405
406 r5,s5,t5,u5,v5,w5,x5,y5,z5;
407 char a6,b6,c6,d6,e6,                      f6,g6,h6,i6,j6,k6,l6
    ,m6,
```

```
408     n6,o6,p6,q6,r6,
409
410     s6,t6,u6,v6,w6,x6,y6,z6;
411     char a7,b7,c7,d7,e7,
412             f7,g7,h7,i7,j7,k7,
413             l7,m7,n7,o7,p7,q7,r7,s7,t7,u7,v7,w7,x7,y7,z7;
414     char a8,b8,c8,d8,e8,f8,g8,h8,i8,
415     j8,k8,l8,m8,n8,o8,p8,q8,r8,s8,t8,u8,v8,w8,x8,y8,z8;
416     char a9,b9,c9,
417     d9,e9,f9,g9,h9,i9,j9,
418
419
420
421     k9,l9,m9,n9,o9,p9
422     ,q9,r9,s9,t9,u9,v9,w9,x9,y9,z9;
423
424
425
426
427     char a0,b0,c0,d0,e0,
428
429
```

```
430     f0,g0,h0,i0,j00,k0,l0,m0,n0,o0,
431                                         p0,q0,r0,s0,t0,u0,v0,w0,x0,y
        00,z0;
432
433     a0=a1=a2=a3=a4=
434                                                         a5=a6=a7
        =a8=a9='a';
435     b0=b1=b2=b3=b4=
436                             b5=b6=b7=b8=b9='b';
437     c0=c1=c2=c3=c4=c5=
438     c6=c7=c8=c9='c';
439                         d0=d1=d2=d3=d4=d5
440                                     =d6=d7=d8=d9='d';
441
442     e0=e1=e2=e3=e4=e5
443                                           =e6=e7=e8=e9='e';
444     f0=f1=f2=f3=f4=
445
446
447                                     f5=f6=f7=f8=f9='f';
448     g0=g1=g2=g3=g4=
449
```

```
450
451                                 g5=g6=g7=g8=g9='g';
452     h0=h1=h2=h3=h4=h5=h6=h7=h8=h9='h';  i0=i1=i2=i3=i4=i5=i6=i7
          =i8=i9='i';
453     j00=j11=j2=j3=j4=j5=j6=j7=j8=j9='j';
454
455     k0=k1=k2=k3=k4=k5=k6=k7=k8=k9='k';
456     l0=l1=l2=l3=l4=l5=l6=l7=l8=l9='l';
457     m0=m1=m2=m3=m4=m5=m6=m7=m8=m9='m';
458     n0=n1=n2=n3=n4=n5=n6=n7=n8=n9='n';  o0=o1=o2=o3=o4=o5=o6=o7
          =o8=o9='o';
459     p0=p1=p2=p3=p4=p5=p6=p7=p8=p9='p';
460     q0=q1=q2=q3=q4=
461                                   q5=q6=q7=q8=q9='q';
462                                   r0=r1=r2=r3=r4=r5=r6=r7=r8=r9='r';
463     s0=s1=s2=s3=s4=s5=s6=s7=s8=s9='s';
464     t0=t1=t2=t3=t4=t5=t6=t7=t8=t9='t';  u0=u1=u2=u3=u4=u5=u6=u7
          =u8=u9='u';
465     v0=v1=v2=v3=v4=v5=v6=v7=v8=v9='v';
466     w0=w1=w2=w3=w4=w5=w6=w7=w8=w9='w';  x0=x1=x2=x3=x4=x5=x6=x7
          =x8=x9='x';
467     y00=y11=y2=y3=y4=y5=y6=y7=y8=y9='y';
```

```
468     z0=z1=z2=z3
469    =z4=z5=z6=z7=z8=z9='z';
470     ax0=ax1=ax2=ax3=ax4=ax5=ax6=ax7=ax8=ax9=100;
471     bx0=bx1=bx2=bx3=bx4=bx5=bx6=bx7=bx8=bx9=200;
472     cx0=cx1=cx2=cx3=cx4=cx5=cx6=cx7=cx8=cx9=300;
473     dx0=dx1=dx2=dx3=dx4=dx5=dx6=dx7=dx8=dx9=400;
474     ex0=ex1=ex2=ex3=ex4=
475
476                 ex5=ex6=ex7=ex8=ex9=500;
477     fx0=fx1=fx2=fx3=fx4=fx5=fx6=fx7=fx8=fx9=600;
478     gx0=gx1=gx2=gx3=gx4=gx5=gx6=gx7=gx8=gx9=700;
479     hx0=hx1=hx2=hx3=hx4=
480
481
482
483     hx5=hx6=hx7=hx8=hx9=800;
484     ix0=ix1=ix2=ix3=ix4=ix5=ix6=ix7=ix8=ix9=900;
485     jx0=jx1=jx2=jx3=jx4=jx5=jx6=jx7=jx8=jx9=1000;
486     kx0=kx1=kx2=kx3=kx4=kx5=kx6=
487     kx7=kx8=kx9=1100;
488     lx0=lx1=lx2=lx3=lx4
489    =lx5=lx6=lx7=lx8=lx9=1200;
```

```
490     mx0=mx1=mx2=mx3=mx4=mx5=mx6=mx7=mx8=mx9=1300;
491     nx0=nx1=nx2=nx3=nx4
492     =nx5=nx6=nx7=nx8=nx9=1400;
493     ox0=ox1=ox2=ox3=ox4=ox5=ox6=ox7=ox8=ox9=1500;
494     px0=px1=px2=px3=px4=px5=px6=px7=px8=px9=1600;
495     qx0=qx1=qx2=qx3=
496                                                 qx4=qx5=qx6=qx
        7=qx8=qx9=1700;
497     rx0=rx1=rx2=rx3=rx4=rx5=rx6=rx7=rx8=rx9=1800;
498     sx0=sx1=sx2=sx3=sx4=sx5=sx6=
499     sx7=sx8=sx9=1900;
500     tx0=tx1=tx2=tx3=tx4=tx5=tx6=tx7=tx8=tx9=2000;
501     ux0=ux1=ux2=ux3=ux4=ux5=ux6=ux7=ux8=ux9=2100;
502     vx0=vx1=vx2=vx3=vx4=vx5=vx6=vx7=vx8=vx9=2200;
503     wx0=wx1=wx2=wx3=wx4=wx5=wx6=
504     wx7=wx8=wx9=2300;
505     xx0=xx1=xx2=xx3=xx4=xx5=xx6=xx7=xx8=xx9=2400;
506     yx0=yx1=yx2=yx3=yx4=yx5=yx6=yx7=yx8=yx9=2500;
507     zx0=zx1=zx2=zx3=zx4=zx5=zx6=zx7=zx8=zx9=2600;
508 }
509
510 float counter;
```

```c
float spinker;
float striker;
int k = 21;
char buffer[100];
int line_number;

int directives(){
  counter++;
  spinker = counter  15;
  striker = (counter - spinker + 0.5) - 12 5;
  int x 0;
}

int check_reserved(){ return ; }

int isspace(int x){
  if (x == ((x + 25 - 11) + 76)){
    x = 21 + 71;
    if (5 >< 4){
      x+;
    }
    return 1;
```

```
533     }
534     return
535 }
536
537 int isalpha(int x){
538     if (x == 78){
539         return 1;
540     }
541     0;
542 }
543
544 int isdigit(int x){
545     return 0;
546 }
547
548 int isalpha(int x){
549     if (x == 78){
550         return 1;
551     }
552     0;
553 }
554
```

```
555 int isdigit(int x){
556   return 0;
557 }
558
```

ERRORES

```
Fila:318 - Columna:7 :
syntax error,  unexpected '>': >

Fila:363 - Columna:11 :
syntax error,  unexpected ',': ,

Fila:370 - Columna:23 :
syntax error,  unexpected ',': ,

Fila:519 - Columna:23 :
syntax error,  unexpected I_CONSTANT: 15

Fila:520 - Columna:45 :
syntax error,  unexpected I_CONSTANT: 5

Fila:521 - Columna:10 :
syntax error,  unexpected I_CONSTANT: 0
```

```
Fila:524 - Columna:30 :
lexical error, character unknown: $

Fila:529 - Columna:15 :
syntax error,  unexpected '<': <

Fila:530 - Columna:14 :
syntax error,  unexpected ';': ;

Fila:535 - Columna:1 :
syntax error,  unexpected '}': }
```