Parite 1:

1)

G=<T,N,S,P>

T={a=[a-z|A-Z], c=[1-9], THEN,BEGIN,ELSE,IF,END.,VAR,CONST,O=[' ‘,’+’,’-’,’*’], PROGRAM}

N={S,S1 ,S1’,S2,S3,S3’,S4,S5,S5’,S5'’,S6,S8,S9,S9,S10,S11,S11'’S11',S12,S12',

S13,S14,

S14',S15,S15’,

Chiffre ,Chiffre'’,

Chiffre',ValS ,ValS',S16,S17,S17’,S18,S19,S19'’,S19',S20,S20',

S21,S22,S8A,

S9A,

S9A',

S10A,

S11A,

S11A'’,

S11A',

S12A,S12A',

S13A ,S14A,

S14A',

S15A,

S15A’,

ChiffreA,

Chiffre'’,

ChiffreA'

ValSA ,

ValSA',

S16A,

S17A,

S17A’,

S18A,

S19A,

S19A'’,

S19A',|

S20A,

S20A',

S21A,

S22

}

P={

S-->PROGRAM S1

S1 --> aS1’

S1’-->S1|cS1’|a;S2|cS2

S2--> conSt S3| var S3| BEGIN S8

S3-->aS3’

S3’ -->aS3’|cS3’|aS4|cS4|s4

S4-->,S3|;S2’|=OS5

S5-->cS5’|.S5’’|S6

S5’-->cS5’|.cS5’’|s6

S5''-->cS5’’|cS6

S6-->,S3|;S2’|;BEGIN S8

S8 -->S9|if(S14

S9-->aS9’

S9'-->S9|cS9’|a=OS10|c=OS10|=OS10

S10=S11|S12

S11-->cS11’|.S11’’

S11''-->cS11’’|cS13

S11'--> cS11’|.S11’’|c;S13

S12-->aS12’

S12'-->S12|cS12’|a;S13|c;S13| S13

S13 --> S8 |END.

S14-->aS14’

S14'-->S14|cS14’|a==S15|c==S16|==S15

S15-->O chiffre |O char

S15’--> chiffre | char

Chiffre --> c chiffre’|. Chiffre''

Chiffre''--> c chiffre’’|c + S15’| c - S15’| c * S15’| c)THEN S16

Chiffre'-->chiffre|c S15|) THEN S16

ValS --> a valS’

ValS'--> a valS’|c valS’| c + S15’| c - S15’| c * S15’|S16

S16-->S17|{S22

S17-->aS17’

S17’-->S17|cS17|a=OS18| c=OS18

S18-->S19|S20

S19-->cS19’|.S19’’

S19'’-->cS19’’|c;S21

S19'-->cS19’|.S19”|c;S21

S20-->aS20’

S20'-->S20|cS20’|c;S21|a;S21

S21-->FSI; S8| FSI END.

S22-->S8A

S8A -->S9A|if(S14A

S9A-->aS9A’

S9A'-->S9A|cS9A’|a=OS10A|c=OS10A

S10A=S11A|S12A

S11A-->cS11A’|.S11A’’

S11A'’-->cS11A’’|cS13A

S11A'--> cS11A’|.S11A’’|c;S13A

S12A-->aS12A’

S12A'-->S12A|cS12A’|a;S13A|c;S13A

S13A --> S8A |ELSE S8A |S21

```
S14A-->aS14A’

S14A'-->S14A|cS14A’|a==S15A|c==S16A

S15A-->O chiffreA  |O charA

S15A’--> chiffreA  | charA


ChiffreA  --> c chiffreA’|.  ChiffreA''

Chiffre''-->  c chiffreA’’|c   + S15A’| c  - S15A’| c  * S15A’| c)THEN S16A

ChiffreA'-->chiffreA|c  S15A| ) THEN S16A

ValSA --> a valSA’

ValSA'--> a valSA’|c valSA’| c  + S15A’| c  - S15A’| c  * S15A’|S16A

S16A-->S17A|{S22A

S17A-->aS17A’

S17A’-->S17A|cS17A|a=OS18A|  c=OS18A

S18A-->S19A|S20A

S19A-->cS19A’|.S19A’’

S19A''-->cS19A’’|c;S21A

S19A'-->cS19A’|.S19A”|c;S21A

S20A-->aS20A’

S20A'-->S20A|cS20A’|c;S21A|a;S21A

S21A-->FSI; S8A|FSI;}END .

S22-->s8A

}
```

Partie 2:

1)

	/	*	%	c
S0	S1	p	p	p
S1	p	S2	p	p
p	p	p	p	p
S2	S6	S3	S5	S2
S3	S4	S3	S5	S2
S5	S2	S2	S5	S2
S6	S6	p	S5	S2
S4	p	p	p	p

2)

S0-->/ S1| c p |* p | % p

S1-->/ p| c p |* S2 | % p

p-->/ p| c p |* p | % p

S2-->/ S6| c S2 |*S3 | % S5

S3-->/ S4| c S2 |* S3 | % S5

S5-->/ S2| c 5 |* S2 | % 2S

S6-->/ S6| c S5 |* p | % S2

S4-->/ p| c p |* p | % p

3)

```
#include <stdio.h>

#include <ctype.h>

#include <stdbool.h>
```

```c
#include <stdbool.h>

#include <string.h>




bool  cinclude(char  c) {

    char tab[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '0', '1', '2',
'3', '4', '5', '6', '7', '8', '9', '<', '>', '=', '(', ')', ' ', ',', ';', '.', '$', '{', '}'};

    int tab_size = sizeof(tab)  / sizeof(tab[0]);

    int i;


    for (i = 0; i < tab_size;  i++) {

        if (c == tab[i]) {

            return true;

        }

    }


    return false;

}


bool s0(const  char *input);

bool s1(const  char *input);

bool p(const  char *input);

bool s2(const  char *input);
bool s3(const char *input);
bool s5(const char *input);
bool s6(const char *input);
bool s4(const char *input);

bool intochar(const char *string) {
  return s0(string);
}

bool s0(const char *input) {
  if (*input == '/') {
input++;
    return s1(input);

  } else {
    return false;
  }
}

bool s1(const char *input) {
  if (*input == '/') {
    input++;
    return p(input);
  } else if (*input == '*') {
    input++;
    return s2(input);
  } else if (*input == '%') {
    input++;
    return p(input);
  } else if (cinclude(*input))  {
    input++;
    return p(input);
  } else {
    return false;
  }
}

bool p(const char *input) {

    return false;

}

bool s2(const char *input) {
  if (*input == '/') {
    input++;
    return s6(input);
  } else if (*input == '*') {
    input++;
    return s3(input);
  } else if (*input == '%') {
    input++;
    return s5(input);
  } else if (cinclude(*input))  {
    input++;
    return s2(input);
  } else {
    return false;
  }
}

bool s3(const char *input) {
  if (*input == '/') {
    input++;
    return s4(input);
  } else if (*input == '*') {
    input++;
    return s3(input);
  } else if (*input == '%') {
    input++;
    return s5(input);
  } else if (cinclude(*input))  {
    input++;
    return s2(input);
  } else {
    return false;
  }
}

bool s5(const char *input) {
  if (*input == '/') {
    input++;
    return s2(input);
  } else if (*input == '*') {
    input++;
    return s2(input);
  } else if (*input == '%') {
    input++;
    return s5(input);
  } else if (cinclude(*input))  {
    input++;
    return s2(input);
  } else {
    return false;
  }
}

bool s6(const char *input) {
  if (*input == '/') {
    input++;
    return s6(input);
  } else if (*input == '*') {
    input++;
    return p(input);
  } else if (*input == '%') {
    input++;
    return s5(input);
  } else if (cinclude(*input))  {
    input++;
    return s2(input);
  } else {
    return false;

  }
}

bool s4(const char *input) {


    return true;
```

```
}
int main() {
   char *string = "/*hfl*/";

  if (intochar(string))  {
    printf("la  commentaire \"%s\"  est valid par le  grammar\n", string);
  } else {
    printf("la  commentaire \"%s\"  n'est pas valid par le  grammar\n",
string);
  }
  return  0;
}
```