Pràctica LP (Compiladors): The Lego Game

Cal fer un compilador per interpretar un llenguatge simple de posicionament de blocs rectangulars de Lego en una graella bidimensional $N \times M$. La posició $(1,1)$ de la graella està a la cantonada superior esquerra, mentre que la (N,M) està a la cantonada inferior dreta. Inicialment, la graella està buida, i assumirem que els blocs tenen la mateixa alçada unitària però diferents dimensions, que es poden definir amb un parell d'enters ordenats que especificaran l'amplada (eix de les abscisses), i la longitud (eix de les ordenades). Els blocs tenen una codificació idèntica a la de la graella. Un bloc es pot posar a la graella o a sobre d'altre bloc, sempre que totes les connexions del bloc a posar o empilar quedin connectades. També es poden moure blocs sobre la graella sempre que no hi hagi cap solapament amb cap altre construcció. Amb l'ànim d'automatitzar el procés, el jugador pot definir bucles i funcions per a anar fent l'emplaçament dels blocs en la graella. Quan una instrucció sobre un bloc no compleix les regles, s'ignora. A continuació podeu veure un exemple complet d'aquest llenguatge.

```
Grid 10 20                                  // Comença el joc Graella NxM
B1 = PLACE (2,2) AT (1,1)                   // Crea el bloc B1 amb dimensions (2,2) a la posicio (1,1)
B2 = PLACE (2,4) AT (3,1)                   // Crea el bloc B2 que es situa just al costat de B1
MOVE B2 WEST 1                              // Incorrecta: solapa B2 amb B1

MOVE B2 EAST 5                              // Mou B2 cinc unitats cap a l'est

B2 = B1 PUSH B2                             // Correcta: possa B1 a sobre de B2
                                            // noteu que no s'especifica la posició exacta, només si hi ha espai;
                                            // si es pot, el posarem el més amunt i a l'esquerra possible
B3 = (2,3) PUSH B2                          // Incorrecta: no hi ha espai al bloc B2 pel nou bloc 2x3

B2 = (2,3) PUSH B1 POP B2                   // Correcte: un cop tret B1 es pot posar un nou bloc 2x3
                                            // PUSH i POP associen a la dreta
                                            // Després del POP B1 passa a estar indefinit

WHILE (FITS(B2,2,1,2)) [                    // Bucle que omple amb blocs (2,1) el segon tram (alçada 2) de B2
  B2 = (2,1) PUSH B2                        // Empila el bloc en un espai lliure de B2
]

HEIGHT(B2)                                  // Retorna 2, ja que B2 te:
                                            //    - A alçada 1, un bloc de 2x4
                                            //    - A alçada 2, un bloc de 2x2, i 2 blocs de 2x1

B4 = PLACE (2,2) AT (5,5)
TOWER10                                     // Crida a la funcio TOWER10


DEF TOWER10
WHILE (HEIGHT(B4) > 1 AND HEIGHT(B4) < 10) [ // Bucle per fer una torre d'alçada 10 amb blocs 2x2
  B4 = (2,2) PUSH B4
]
ENDEF
```

Assumiu que com a condicions dels bucles només poden aparèixer els operadors relacionals (<, >), que poden ser combinats amb l'operador `AND`. Fixeu-vos que l'operació `MOVE` actua com assignació, és a dir modifica la definició del block que es mou (i no pot haver cap solapament total o parcial amb cap altre construcció de la graella). El `PUSH` i el `POP` associen a la dreta.

Part 1: Defineix la part lèxica i sintàctica. Fès la gramàtica per a que PCCTS pugui reconèixer-la i decorar-la per generar l'AST mostrat a les pàgines 3 i 4.

Part 2: Interpretació: fés el mètode

```
void executeListInstrucctions(AST *a)
```

que executa la llista d'instruccions apuntada pel paràmetre.

Per a la implementació, pot ser útil definir estructures de dades com ara:

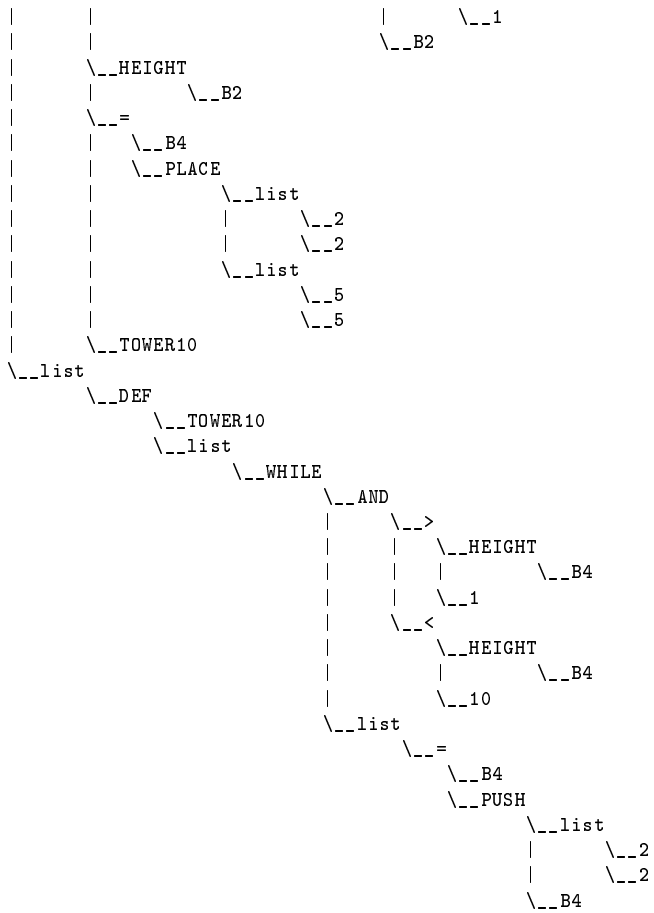
```
typedef struct {
  int x, y;  // punt que determina la cantonada superior esquerra
  int h,w;   // dimensions
} tblock;

typedef struct {
  int n, m;
  vector< vector<int> > height;
  map<string, tblock> blocks;
} Graella;

Graella g;
```

entre d'altres, on `g` és global i inicialment tota la graella està buida i per tant la matriu `height` està tota a zeros. Després del programa anterior, algunes posicions de la graella tindran un valor positiu que indica l'alçada.

```
list
 \__Grid
 |       \__10
 |       \__20
 \__list
 |       \__=
 |       |   \__B1
 |       |   \__PLACE
 |       |           \__list
 |       |           |       \__2
 |       |           |       \__2
 |       |           \__list
 |       |                   \__1
 |       |                   \__1
 |       \__=
 |       |   \__B2
 |       |   \__PLACE
 |       |           \__list
 |       |           |       \__2
 |       |           |       \__4
 |       |           \__list
 |       |                   \__3
 |       |                   \__1
 |       \__MOVE
 |       |       \__B2
 |       |       \__WEST
 |       |       \__1
 |       \__MOVE
 |       |       \__B2
 |       |       \__EAST
 |       |       \__5
 |       \__=
 |       |   \__B2
 |       |   \__PUSH
 |       |           \__B1
 |       |           \__B2
 |       \__=
 |       |   \__B3
 |       |   \__PUSH
 |       |           \__list
 |       |           |       \__2
 |       |           |       \__3
 |       |           \__B2
 |       \__=
 |       |   \__B2
 |       |   \__PUSH
 |       |           \__list
 |       |           |       \__2
 |       |           |       \__3
 |       |           \__POP
 |       |                   \__B1
 |       |                   \__B2
 |       \__WHILE
 |       |       \__FITS
 |       |       |       \__B2
 |       |       |       \__list
 |       |       |       |       \__2
 |       |       |       |       \__1
 |       |       |       \__2
 |       |       \__list
 |       |               \__=
 |       |                   \__B2
 |       |                   \__PUSH
 |       |                           \__list
 |       |                           |       \__2
```


| | | __1
| | __B2
| __HEIGHT
| | __B2
| __=
| | __B4
| | __PLACE
| | __list
| | | __2
| | | __2
| | __list
| | __5
| | __5
| __TOWER10
__list
       __DEF
            __TOWER10
            __list
                  __WHILE
                         __AND
                         | __>
                         | | __HEIGHT
                         | | | __B4
                         | | __1
                         | __<
                         | __HEIGHT
                         | | __B4
                         | __10
                         __list
                                __=
                                    __B4
                                    __PUSH
                                           __list
                                           | __2
                                           | __2
                                           __B4