

#### Devoir 04 :

Nous fournissons un programme qui génère un labyrinthe de taille quelconque sous la forme de lignes de texte. La première ligne indique la largeur du labyrinthe, la seconde sa hauteur, et les lignes qui suivent représentent le labyrinthe à proprement parler où les murs sont donnés par une étoile “\*” et les positions libres par un espace “ ”. Enfin le programme indique une position de départ “D” et une d’arrivée “A”.

Vous devez écrire le programme lisant le labyrinthe généré (sortie du programme cité plus haut) et produisant la liste des positions successives allant du point de départ au point d’arrivée.

Ci-après un exemple de labyrinthe généré (cf. figure 1), suivi de la sortie (de votre programme) attendue (cf. figure 2) :

```

1 21
2 31
3 *****
4 *
5 *****
6 * * * * *
7 * * * * *
8 * * * * *
9 * * * * *
10 * * * * *
11 * * * * *
12 * * * * *
13 * * * * *
14 * * * * *
15 * * * * *
16 * * * * *
17 * * * * *
18 * * * * *
19 * * * * *
20 * * * * *
21 * * * * *
22 * D * * *
23 * * * * *
24 * * * * *
25 * * * * *
26 * * * * A *
27 * * * * *
28 * * * * *
29 * * * * *
30 * * * * *
31 * * * * *
32 * * * * *
33 *****

```

FIGURE 1 – Labyrinthe généré (entrée de votre programme).

```

1  (3,19)
2  (3,20)
3  (3,21)
4  (3,22)
5  (3,23)
6  (4,23)
7  (5,23)
8  (5,24)
9  (5,25)
10 (6,25)
11 (7,25)
12 (8,25)
13 (9,25)
14 (9,26)
15 (9,27)
16 (10,27)
17 (11,27)
18 (12,27)
19 (13,27)
20 (13,28)
21 (13,29)
22 (14,29)
23 (15,29)
24 (15,28)
25 (15,27)
26 (15,26)
27 (15,25)
28 (16,25)
29 (17,25)
30 (18,25)
31 (19,25)
32 (19,24)
33 (19,23)
34 (18,23)
35 (17,23)

```

FIGURE 2 – **Sortie attendue.**

Un exemple partiel de *parser* est donné en figure 3 pour récupérer l'entrée standard (envoyer l'entrée via *pipe*).

```

1  static int * labyParse(void) {
2      char buf[1024], * s;
3      int *labyrinthe = NULL, w = 0, h = 0, x, y;
4      if(fscanf(stdin, "%d\n", &w) != 1 || fscanf(stdin, "%d\n", &h) != 1)
5          return NULL;
6      /* ... */
7      y = 0;
8      /* allouer la mémoire */
9      while(fgets(buf, sizeof buf, stdin)) {
10         x = 0;
11         for(s = buf, x = 0; *s && *s != '\n'; s++, x++) {
12             /* remplir labyrinthe et fixer le point de départ et le point
13              * d'arrivée */
14         }
15         y++;
16     }
17     return labyrinthe;
18 }

```

FIGURE 3 – **Le parseur d'entrée, à compléter.**