Module Braid_Print

1. On travaille sur les tresses représentées par les mots de tresses sur l'alphabet dans la présentation d'Artin

```
open Braid
```

2. Affiche une représentation ASCII de la tresse fournie

```
let ascii b =
    let print_vert () =
        print_string "␣|" in
    let print_sigma positif = function
        | 1 → print_string "␣\\␣/"
        | 2 → if positif then print_string "␣␣/␣"
                         else print_string "␣␣\\␣"
        | 3 → print_string "␣/␣\\"
        | _ → print_string "␣|␣|" in
    let rec aux = function
        | [] → print_newline ()
        | x :: xs → for l = 1 to 4 do
                        for i = 1 to (x − 1) do
                          print_vert ()
                        done;
                        print_sigma (x > 0) l;
                        for i = x + 2 to b.size do
                          print_vert ()
                        done;
                        print_newline ();
                    done;
                    aux xs in
    aux b.word
```

3. Représentation graphique grâce au module Graphics
Ratio hauteur du sigma / épaisseur des brins

```
let ratio = 10
```

Dessine le croisement sigma positif à la position x,y de hauteur h (coordonnées en bas à gauche de la tresse)

```
let sigma_p x y h = let e = h/ratio in
                    let open Graphics in
                    fill_poly [| (x+h/2−e, y+h); (x+h/2, y+h); (x+h+h/2, y); (x+
```

```
h − e + h/2, y) |];
                    set_color white;
                    fill_poly [| (x+h/2−e, y+e); (x+h/2−e, y); (x+h/2+e, y); (x+
h + h/2, y + h − e);
                                        (x + h + h/2, y + h); (x + h + h/2 − 2 × e, y + h) |];
                    set_color black;
                    fill_poly [| (x + h/2 − e, y); (x + h/2, y); (x + h + h/2, y + h); (x +
h + h/2 − e, y + h) |]
```

Dessine le croisement sigma négatif à la position x,y de hauteur h (coordonnées en bas à gauche de la tresse)

```
let sigma_n x y h = let e = h/ratio in
                    let open Graphics in
                    fill_poly [| (x+h/2−e, y); (x+h/2, y); (x+h+h/2, y + h); (x+
h + h/2 − e, y + h) |];
                    set_color white;
                    fill_poly [| (x + h/2 − e, y + h − e); (x + h/2 − e, y + h); (x +
h/2 + e, y + h); (x + h + h/2, y + e);
                                        (x + h + h/2, y); (x + h + h/2 − 2 × e, y) |];
                    set_color black;
                    fill_poly [| (x+h/2−e, y+h); (x+h/2, y+h); (x+h+h/2, y); (x+
h + h/2 − e, y) |]
```

Dessine une barre verticale (brin ne se croisant pas)

```
let vert x y h = let e = h/ratio in
                 let open Graphics in
                 fill_poly [| (x + h/2 − e, y); (x + h/2, y); (x + h/2, y + h); (x + h/2 −
e, y + h) |]
```

Affichage graphique de la tresse

```
let display b =
    let size = b.size and height = List.length b.word in
    let h = (if height = 0 then 60 else min (600/height) 60) in
    Graphics.open_graph ("␣" ^ (string_of_int (h × size + 20)) ^ "x" ^ (string_of_int (h ×
height + 20)));
    let rec aux x y = function
        | [] → ()
        | g :: s → for i = 0 to ((abs g) − 2) do
                       vert (x + i × h) y h
                   done;
                   (if g > 0 then sigma_p else sigma_n) (x + (((abs g) − 1) × h)) y h;
```

```
                for i = (abs g) + 1 to size do
                    vert (x + i × h) y h
                done;
                aux x (y − h) s in
aux 10 ((height − 1) × h + 10) b.word;
Graphics.read_key (); ()
```