Configuring | Vertical Circulation (3 Wrangles)

Part 1 | Stair Shafts

```
//Set every point in a straight line above or below the function centres to Circulation
//Inputs: Point Cloud and stored Function Centres
//Outputs: Vertical circulation shafts connecting important functions to the ground floor

//Living Unit
int liv [] = expandpointgroup(0, "Living_Centre");

vector liv_v = pointattrib(0, "P", liv[0], 1);
if (rint(@P.x*100)/100 == rint(liv_v.x*100)/100 && rint(@P.z*100)/100 == rint(liv_v.z*100)/100){
    i@Circ = 1;
    i@func_id = 10;
}

//Library
int lib [] = expandpointgroup(0, "Library_Centre");
vector lib_v = pointattrib(0, "P", lib[0], 1);
if (rint(@P.x*100)/100 == rint(lib_v.x*100)/100 && rint(@P.z*100)/100 == rint(lib_v.z*100)/100){
    i@Circ = 1;
    i@func_id = 10;
}

//Restaurant
int rest [] = expandpointgroup(0, "Restaurant_Centre");
vector rest_v = pointattrib(0, "P", rest[0], 1);
if (rint(@P.x*100)/100 == rint(rest_v.x*100)/100 && rint(@P.z*100)/100 == rint(rest_v.z*100)/100){
    i@Circ = 1;
    i@func_id = 10;
}
```

Part 2 | Staircase Access

```
//If there is one set Access to "True" and store the Access Point
//If not check if Function Centres are near and on the same Level
//If it finds a point close to it and that function does not have an Access Point yet
//set Access to "True" and store the point

//Input: Point cloud, stored function centres and circulation shafts
//Output: If possible, access point to the existing circulation shafts for each function

//Getting the function centres
vector liv = detailattrib(0,"Living_Centre",0,1);
vector cin = detailattrib(0,"Cinema_Centre",0,1);
vector rest = detailattrib(0,"Restaurant_Centre",0,1);
vector shop = detailattrib(0,"Shops_Centre",0,1);
vector exh = detailattrib(0,"Exhibition_Centre",0,1);
vector lib = detailattrib(0,"Library_Centre",0,1);
vector sta = detailattrib(0,"Studio_Arts_Centre",0,1);
vector cw = detailattrib(0,"Co_Working_Centre",0,1);
vector gym = detailattrib(0,"Gym_Centre",0,1);

//Getting Id for function centres
int liv_id [] = expandpointgroup(0,"Living_Centre");
int cin_id [] = expandpointgroup(0,"Cinema_Centre");
int rest_id [] = expandpointgroup(0,"Restaurant_Centre");
int shop_id [] = expandpointgroup(0,"Shops_Centre");
int exh_id [] = expandpointgroup(0,"Exhibition_Centre");
int lib_id [] = expandpointgroup(0,"Library_Centre");
int sta_id [] = expandpointgroup(0,"Studio_Arts_Centre");
int cw_id [] = expandpointgroup(0,"Co_Working_Centre");
int gym_id [] = expandpointgroup(0,"Gym_Centre");

//Storing Function Values in Arrays
vector centres [] = array(liv, cin, rest, shop, exh, lib, sta, cw, gym);
int centres_id [] = array(liv_id[0], cin_id[0], rest_id[0], shop_id[0], exh_id[0], lib_id[0], sta_id[0], cw_id[0], gym_id[0]);
string detail [] = {"Liv_Stair", "Cin_Stair", "Rest_Stair", "Shop_Stair", "Exh_Stair", "Lib_Stair", "Sta_Stair", "Cw_Stair", "Gym_Stair"};
string group [] = {"Liv_Acc", "Cin_Acc", "Rest_Acc", "Shop_Acc", "Exh_Acc", "Lib_Acc", "Sta_Acc", "Cw_Acc", "Gym_Acc"};


for(int i = 0; i < 9; i++){

    //Checking for circulation access within own function
    if(@func_id == (i + 1) && rint(@P.y * 100)/100 == rint(centres[i].y *100)/100){
        setdetailattrib(0, detail[i], 1, "set");
        setpointgroup(0, group[i], @ptnum, 1, "set");
    }

    //Checking for circulation access within close proximity of the function centre
    int stair = detailattrib(0,detail[i],0,1);

    if (stair == 0){
        int pts[] = nearpoints(0, @P, 15); //Searching for Points around Stairs
        int test  = find(pts, centres_id[i]);

        if(test > -1 && rint(@P.y * 100)/100 == rint(centres[i].y *100)/100){
            setdetailattrib(0, detail[i], 1, "set");
            setpointgroup(0, group[i], @ptnum, 1, "set");
        }
    }
}
```

Part 3 | Complementary Stairs

```
//If the function does not have any access to vertical circulation
//Set every point in a straight line above and below the function centre as circulation
//Creating another Circulation Shaft

//Input: Point cloud, stored function centres and stored circulation shafts access
//Output: Vertical circulation shafts for every function not having ground floor access yet

//Getting the function centres
vector liv = detailattrib(0,"Living_Centre",0,1);
vector cin = detailattrib(0,"Cinema_Centre",0,1);
vector rest = detailattrib(0,"Restaurant_Centre",0,1);
vector shop = detailattrib(0,"Shops_Centre",0,1);
vector exh = detailattrib(0,"Exhibition_Centre",0,1);
vector lib = detailattrib(0,"Library_Centre",0,1);
vector sta = detailattrib(0,"Studio_Arts_Centre",0,1);
vector cw = detailattrib(0,"Co_Working_Centre",0,1);
vector gym = detailattrib(0,"Gym_Centre",0,1);

//Storing Function Values in Arrays
vector centres [] = array(liv, cin, rest, shop, exh, lib, sta, cw, gym);
string detail [] = {"Liv_Stair", "Cin_Stair", "Rest_Stair", "Shop_Stair", "Exh_Stair", "Lib_Stair", "Sta_Stair", "Cw_Stair", "Gym_Stair"};

//Assigning Circulation Attribute and Group to Points
for(int i = 0; i < 9; i++){
  int stair = detailattrib(0,detail[i],0,1);

  if(stair == 0){
    if (rint(@P.x*100)/100 == rint(centres[i].x*100)/100 && rint(@P.z*100)/100 == rint(centres[i].z*100)/100){
      i@Circ = 1;
      i@func_id = 10;
    }
  }
}
```