**TD10 – C**

**Exercice 1 :**

On commence à l’@ : 0x100 et sizeof(int)=4

int a = 1;

int b = 2;

int\* pt=&b;

int\* pt2=&a;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x100 | 0x104 | | 0x108 | | 0x10C | | 0x110 |
| 1 | | 2 | | 0x104 | | 0x100 | |

int\* pt=&b; 🡪 0x100

int\* pt2=&a; 🡪 0x104

int \* c = (int\*)calloc(5,sizeof(int));

\*c = a;

\*(c+1) = \*pt2;

c[1] = 5;

c[2] = 10;

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x110 | 0x114 | | 0x118 | | 0x11C | | 0x120 | | 0x124 |
| 1 | | 5 | | 10 | | 0 | | 0 | |

float f[3] = {3.5, 2, 10.1}; //sizeof(float) = 8

void \* pt3 = &f+1;

char \* mot = "bonjour"

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x124 | 0x12C | | 0x134 | | 0x13D | | 0x142 | | 0x14C | | 0x150 | |  | |  | |  | |  | |  | |  |
| 3,5 | | 2 | | 10.1 | | 0x128 | | B | | O | | N | | J | | O | | U | | R | | \n | |

**Exercice 2 :**

Fonction Tri(tableau d’entier liste, entier longueur (du tableau)){

Entier compteur = 0

Entier swap = 0 ;

Pour(entier i = 0, tant que i < entier longueur -1 ; entier i ++){

Si(liste[i] > liste[i+1]){

Swap = liste[i]

liste[i] = liste[i + 1];

liste[i + 1] = swap;

compteur++;

}

Si (compteur != 0)

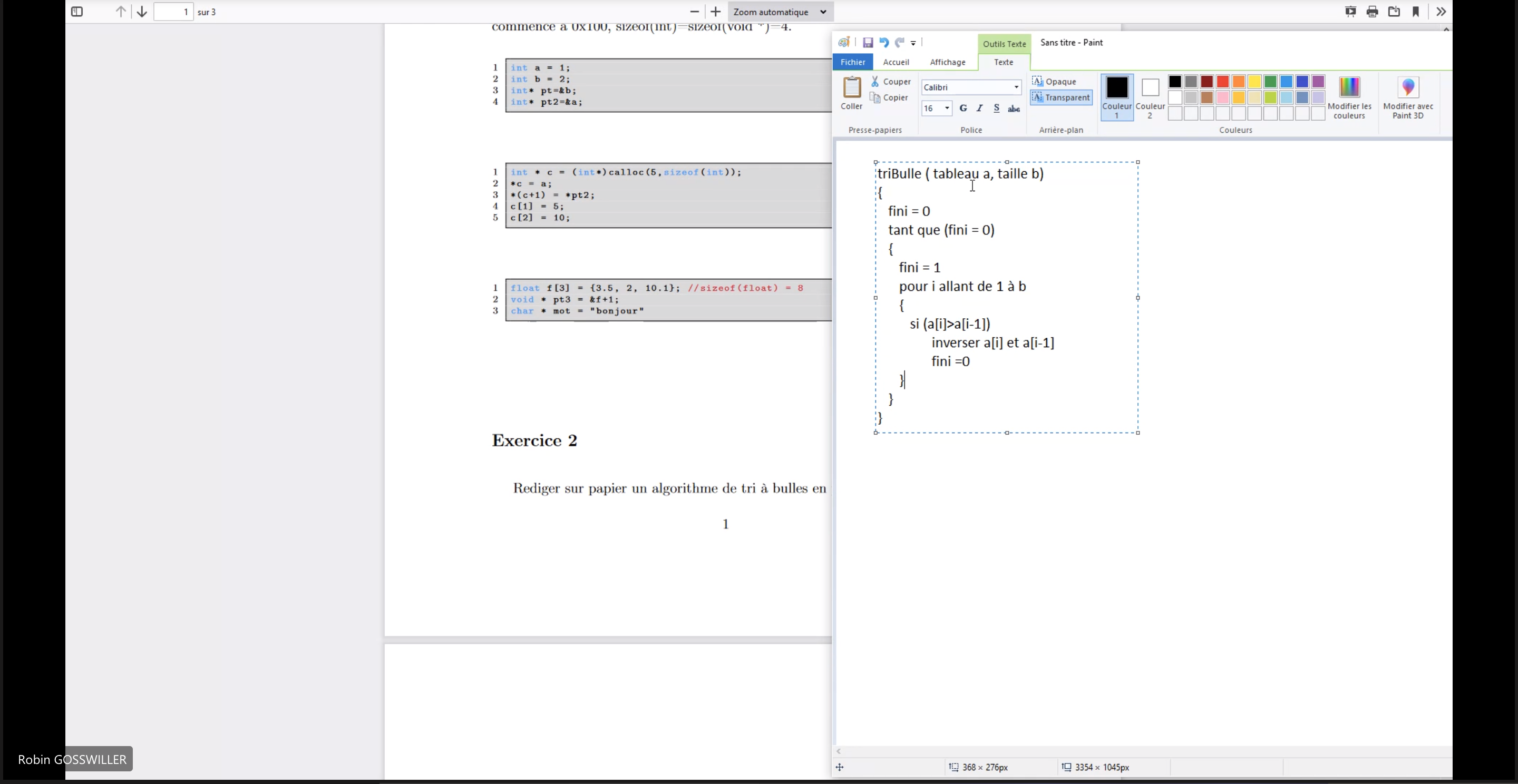
{

Réexcute la fonction Tri (compteur, longueur)

}

}

}



**Exercice 4 :**

typedef struct node\_ {

int value;

struct node\_\* gauche;

struct node\_\* droite;

} node;

typedef node\* pNode;

pNode newNode(int a){

pNode n = (pNode)malloc(sizeof(struct node\_));

n->value = a;

n->droite = NULL;

n->gauche = NULL;

return n;

}

//les fonctions addNodeLeft et addNodeRight ajoutent une feuille

//respectivement a gauche et a droite d’une racine

pNode root = newNode (10);

pNode cinq = addNodeLeft(5, root);

pNode deux = addNodeLeft(2, cinq);

pNode vingt = addNodeRight(20, root);

pNode quinze = addNodeLeft(15, vingt);

Une image contenant texte

Description générée automatiquement