Travail Pratique : Calcul du PGCD avec l’algorithme d’Euclide

Tâche de travail

• Introduction

• Rappel théorique

• Exemple manuel

• Implémentation

• Vérification

• Conclusion

⸻

Introduction

Dans le cadre de ce travail pratique, il s’agit d’analyser et de mettre en œuvre l’algorithme d’Euclide afin de calculer le plus grand commun diviseur (PGCD) de deux entiers naturels.

⸻

Rappel théorique

Le PGCD (Plus Grand Commun Diviseur) de deux nombres est le plus grand entier qui divise simultanément ces deux nombres sans laisser de reste.

L’algorithme d’Euclide repose sur le principe suivant :

Le PGCD de deux nombres a et b (avec a > b) est égal au PGCD de b et du reste de la division de a par b.

Formule :

PGCD(a, b) = PGCD(b, a % b)

jusqu’à ce que b = 0, alors PGCD(a, 0) = a

⸻

## Exemple manuel

Calculons le PGCD de 48 et 18 :

Etape a b a%b

1 48 18 12

2 18 12 6

3 12 6 0

👉 PGCD(48, 18) = 6

⸻

Implémentation en langage C

#include <stdio.h>

int main() {

int a, b, reste;

printf("Entrez deux entiers : ");

scanf("%d %d", &a, &b);

while (b != 0) {

reste = a % b;

a = b;

b = reste;

}

printf("Le PGCD est : %d\n", a);

return 0;

}

Vérification

Exemple d’exécution :

Entrée : 48 18

Sortie : Le PGCD est : 6

Test supplémentaire :

Entrée : 81 27

Sortie : Le PGCD est : 27

⸻

# Conclusion

L’algorithme d’Euclide est une méthode simple et efficace pour déterminer le PGCD de deux entiers.

Il repose sur un principe de divisions successives et peut être facilement implémenté dans n’importe quel langage de programmation, notamment en C.

#include <stdio.h>

// Fonction pour calculer le PGCD et les coefficients de Bézout

int bezout(int a, int b, int \*u, int \*v) {

if (b == 0) {

\*u = 1;

\*v = 0;

return a;

} else {

int u1, v1;

int d = bezout(b, a % b, &u1, &v1);

\*u = v1;

\*v = u1 - (a / b) \* v1;

return d;

}

}

int main() {

int a, b, r;

printf("Entrez deux entiers positifs: ");

scanf("%d %d", &a, &b);

printf("Étapes de l'algorithme d'Euclide:\n");

int x = a, y = b; // garder les valeurs originales

while (b!= 0) {

r = a % b;

printf("%d = %d \* (%d) + %d\n", a, a / b, b, r);

a = b;

b = r;

}

printf("Le PGCD est: %d\n", a);

if (a == 1) {

printf("Les nombres %d et %d sont premiers entre eux.\n", x, y);

} else {

printf("Les nombres %d et %d ne sont pas premiers entre eux.\n", x, y);

}

// Calcul des coefficients de Bézout

int u, v;

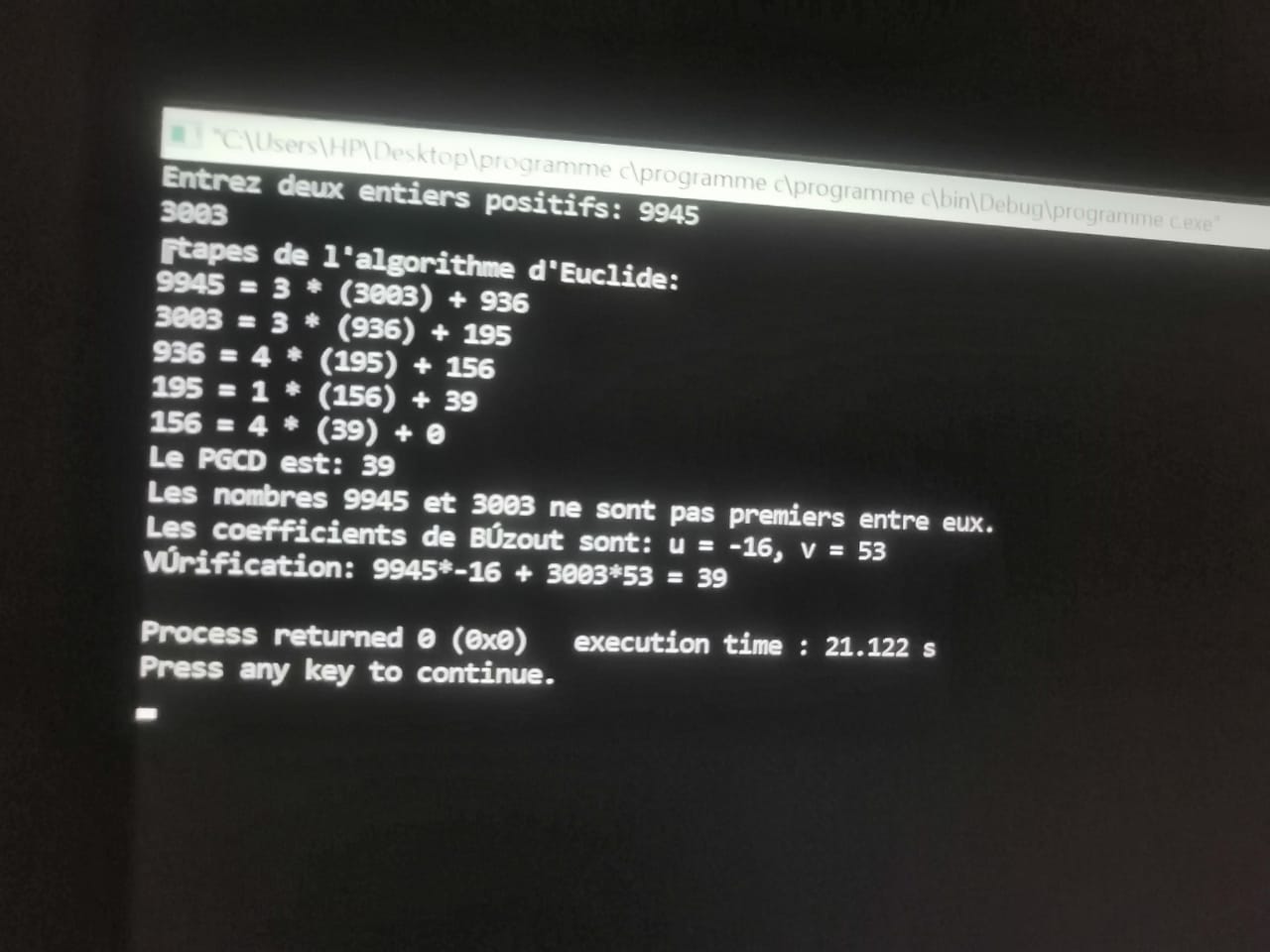
int pgcd = bezout(x, y, &u, &v);

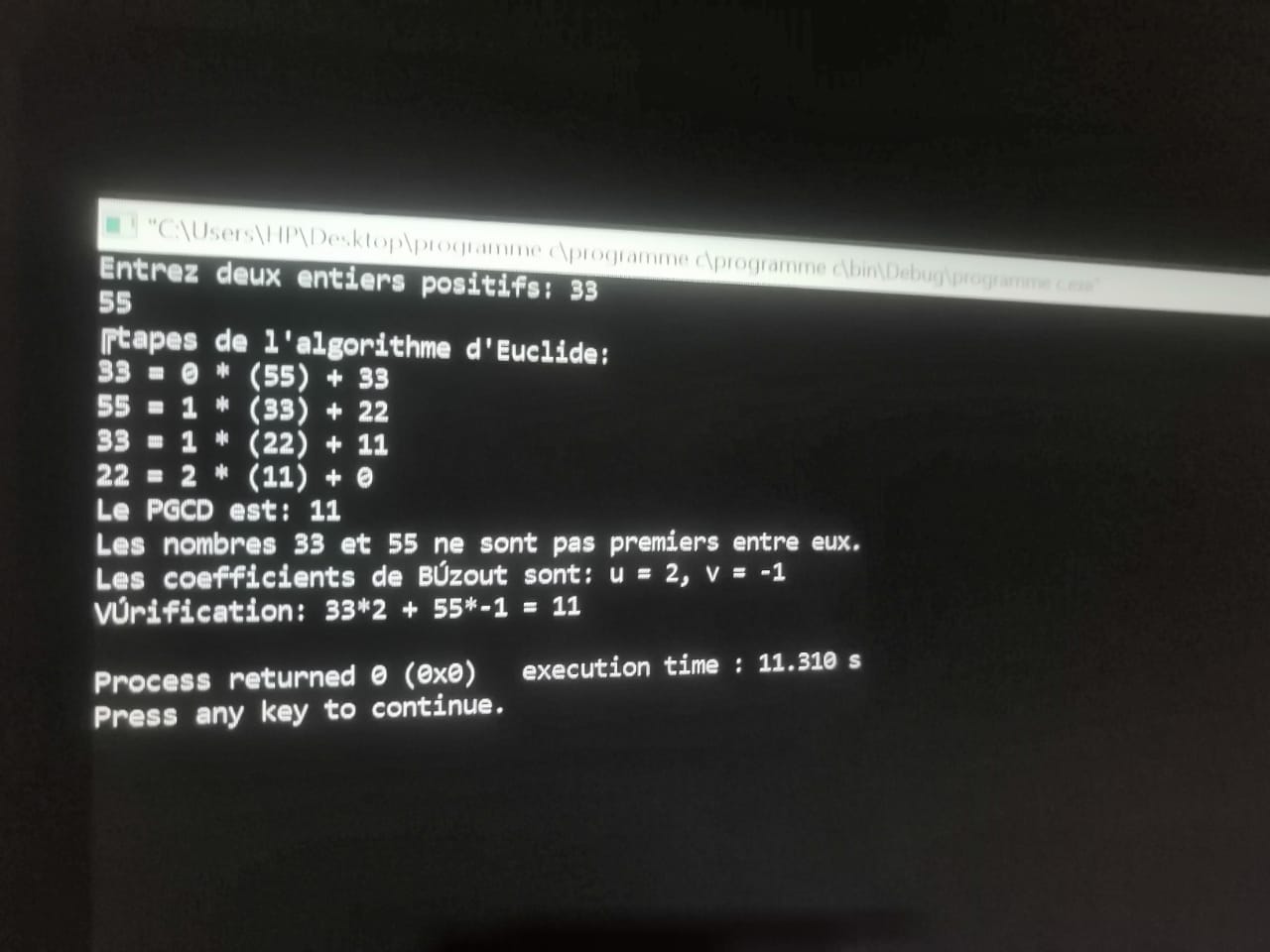
printf("Les coefficients de Bézout sont: u = %d, v = %d\n", u, v);

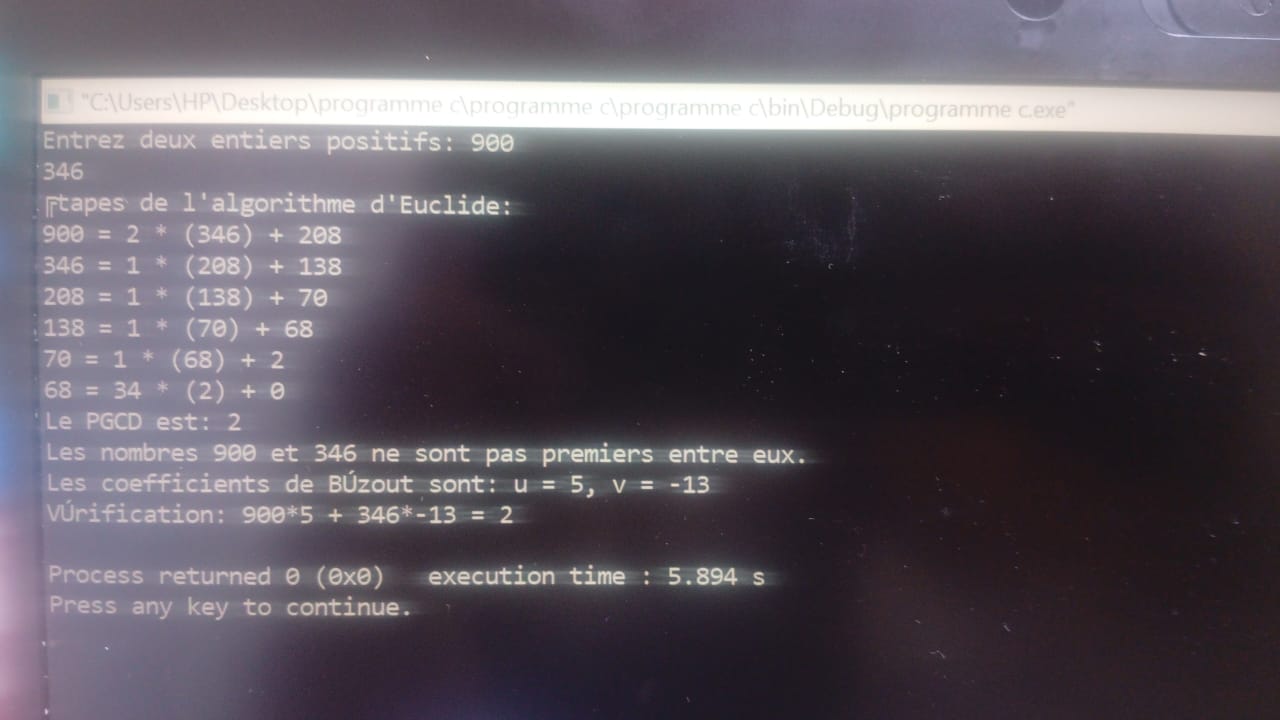
printf("Vérification: %d\*%d + %d\*%d = %d\n", x, u, y, v, x\*u + y\*v);

return 0;

}







# Conclusion

Ce programme illustre efficacement l’algorithme d’Euclide pour le calcul du PGCD (Plus Grand Commun Diviseur) de deux entiers positifs, ainsi que l’application de l’identité de Bézout.

Grâce à une approche récursive, il permet non seulement de déterminer le PGCD, mais aussi les coefficients de Bézout (u et v) vérifiant la relation :

## a \times u + b \times v = \text{PGCD}(a, b)

Cette expérience montre l’importance de l’algorithme d’Euclide dans les calculs arithmétiques et son utilité dans plusieurs domaines, notamment la cryptographie, la résolution d’équations diophantiennes et la théorie des nombres.

En conclusion, ce travail nous a permis de comprendre à la fois le fonctionnement logique d’un algorithme récursif et son implémentation pratique en langage C.