UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et génie informatique

RAPPORT APP2

Introduction à la programmation et aux algorithmes
GEN 146

Présenté à
Équipe de formateurs de la session S1

Présenté par
Raphael Bouchard – bour0703
Alexis Guérard – guea0902

Sherbrooke – 4 octobre 2022

TABLE DES MATIÈRES

1. INTRODUCTION

Une bonne pratique à avoir en étant un ingénieur informatique est de penser à comment créer un code plutôt que de directement commencer à l'écrire. Ce rapport montre donc les différents pseudocode et diagrammes d'activités UML qui ont mené à l'élaboration de différents codes d'opérations mathématiques en langage C dans l'environnement de travail Geany. On retrouve entre autres la recherche de caractère, la détection de palindrome, le calcul du cosinus et du sinus à l'aide de séries et l'addition et la multiplication de matrices [1]. Les différents plans de test sont également montrés.

2. DÉVELOPPEMENT

2.1 RECHERCHE DE CARACTÈRE

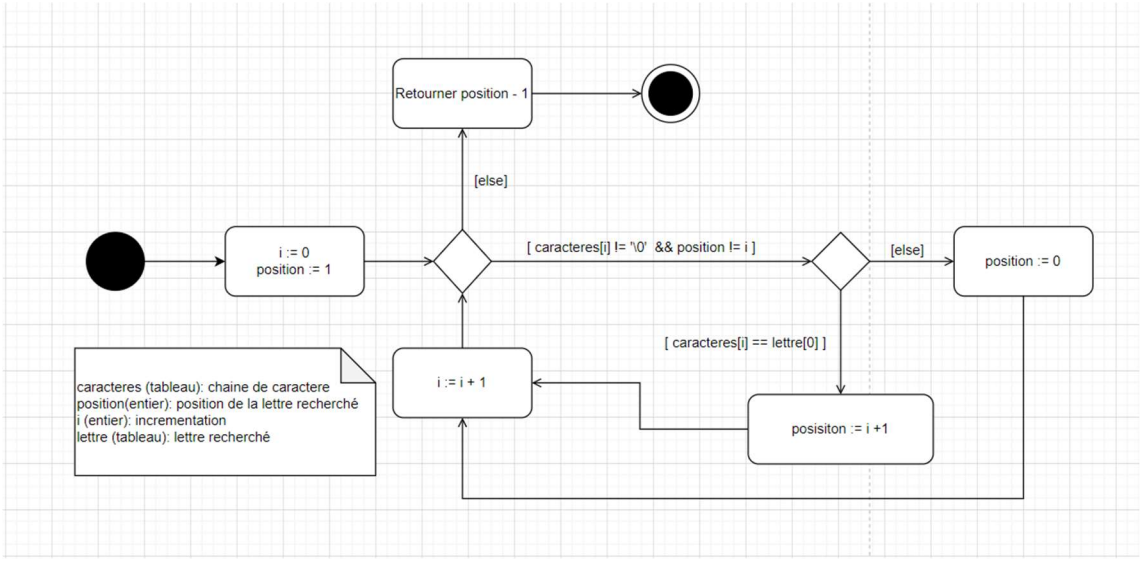

Figure 1 : Diagramme de fonction pour la recherche de caractère

```c
int recherche(char caractere[1024], char lettre[1])
{
    int position = 1;
    for (int i =0; caractere[i]!= '\0' && position != i; i++)
    {
        if (caractere[i] == lettre[0])
        {
            position = i + 1;
        }
        else
        {
            position = 0;
        }
    }
    return position - 1;
}
```

Figure 2 : Code en langage C pour la recherche de caractère

2.2 DÉTECTION DE PALINDROME

Voici le pseudo code et la version en langage C de la fonction de détection de palindrome.

FONCTION LongueurChaine(chaine) : longueur
//Trouve la longueur de la chaine de caractère
//chaine(tableau) : chaine de caractère
//longueur(entier) : longueur de la chaine
DEBUT
// i (entier) : compteur incrémental
i := 0
Longueur := 0
TANT QUE chaine[i] != '\0'
i := i + 1
longueur := longueur + 1
Retourner longueur

FONCTION palindrome(caractères) : validation
//Retourne 1 si la chaine est un palindrome et 0 si elle ne l'est pas
//caractères(tableau) : chaine de caractère
//validation(entier) : validation ou non du palindrome
DEBUT
//début(entier) : compteur incrémentale
//fin(entier) : compteur incrémentale
//b(entier) : compteur incrémentale
fin = LongueurChaine(caractères)
début = 0
b = fin
TANT QUE début != b/2

```c
int longueur(char caractere[1024])
{
    int compteur = 0;
    while (caractere[compteur] != '\0')
    {
        compteur = compteur + 1;
    }
    return compteur;
}
```

Figure 3 : Code en langage C pour trouver la longueur d'une chaine de caractère.

```c
int palindrome(char caractere[1024])
{
    int validation = 1, debut = 0;
    int fin = longueur(caractere)-1, b = fin;
    while (debut != b/2)
    {
        if (caractere[debut] != caractere[fin])
        {
            validation = 0;
        }
        debut++;
        fin--;

    }
    return validation;
}
```

Figure 4 : Code en langage C pour trouver la longueur d'une chaine de caractère.

```
        SI caractères[début] != caractères[fin]
            validation = 0
        début++
        fin—
    Retourne validation
FIN
```

2.3 CALCUL DU SINUS AVEC UNE SÉRIE

Voici le pseudo code et la version en langage c de la fonction du calcul du sinus.

```
FONCTION Factorielle(n) : valeur
        // Calculer la factorielle d'un nombre
        // n(entier) : nombre pour faire la factorielle
        //valeur(entier) : résultat de la factorielle
DEBUT
    valeur := 1
    TANT QUE n > 0
        valeur := valeur * n
        n := n – 1
    Retourner valeur
FIN

FONCTION Puissance(x,n) : valeur
        //Calculer la puissance d'un nombre
        //x (réelle positif) : nombre pour faire le calcul de la puissance
        //n (entier) : exposant pour le calcul
        //valeur (réelle positif) : résultat de la puissance
```

```c
float factorielle(unsigned int n)
{
    float valeur = 1;
    while (n > 0)
    {
        valeur = valeur * n;
        n = n - 1;
    }
    return valeur;
}
```

Figure 5 : Code en langage C pour trouver la factorielle.

```c
float puissance(unsigned int n, float x)
{
    float p = 1;
    while (n > 0)
    {
        p = p * x;
        n = n - 1;
    }
    return p;
}
```

Figure 6 : Code en langage C pour trouver la puissance.

DEBUT

valeur = 1

TANT QUE n > 0

valeur = valeur * x

n = n - 1

Retourner valeur

FIN

FONCTION ApproxSinus(angle) : valeur

//angle (réelle) : angle pour effectuer le sinus

//valeur (réelle) : résultat du sinus

DEBUT

//terme (entier) : nombre de terme pour la série

// i (entier) : compteur incrémentale

// a (entier) : variable pour la puissance et la factorielle

// z (entier) : variable pour l'addition et la soustraction

POUR i = 1 **A** n

valeur = valeur + z * (puissance(angle,a)/(factorielle(a))

a= a + 2

z = z * -1

Retourner valeur

FIN

```c
float ApproxSinus(float angle)
{
    #define terme 5
    float valeur = 0;
    float a = 1, z = 1,i;
    while (angle > pi)
    {
        angle = angle - 2*pi;
    }
    while (angle < -pi)
    {
        angle = angle + 2*pi;
    }

    for (i = 1; i <= terme; i = i + 1)
    {
        valeur = valeur + z*((puissance(a,angle))/(factorielle(a)));
        a = a + 2;
        z = z * -1;
    }
    return valeur;
}
```

Figure 7: Code en langage C pour trouver le sinus d'un angle

2.4 CALCUL DU COSINUS AVEC UNE SÉRIE

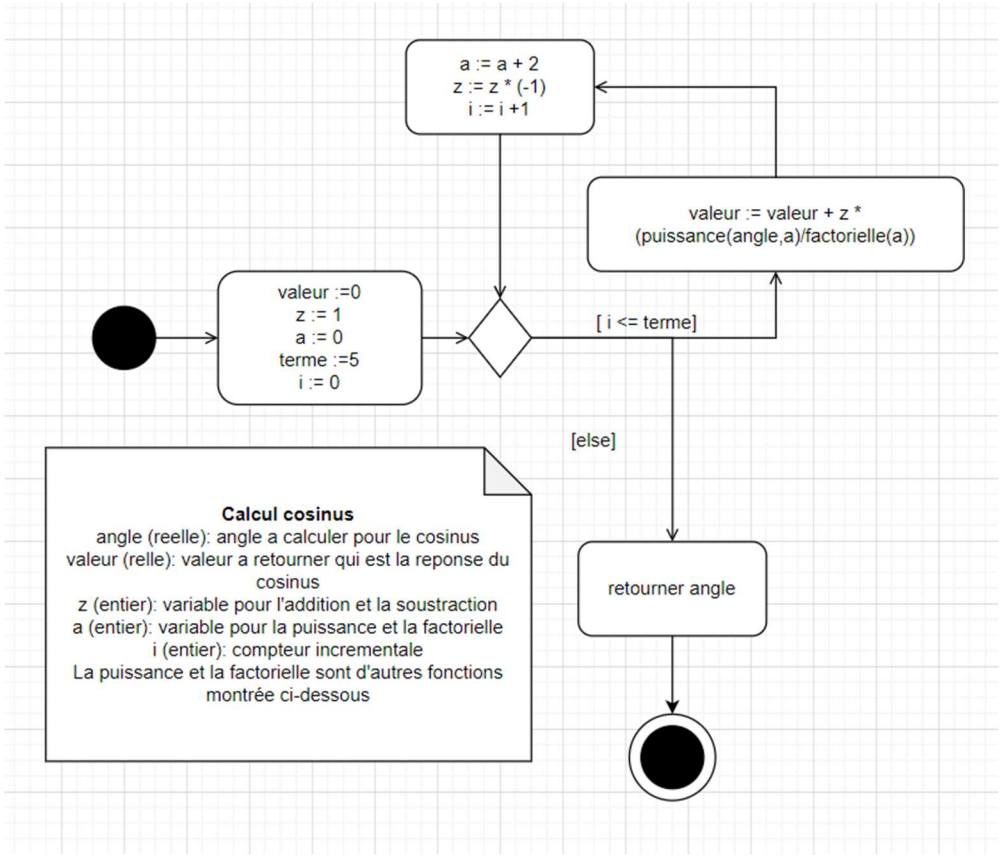

Figure 8: Diagramme de fonctions pour trouver le sinus d'un angle.

```c
float ApproxCosinus(float angle)
{
    #define terme 5
    float valeur = 0;
    float a = 0, z = 1,i;
    while (angle > pi)
    {
        angle = angle - 2*pi;
    }
    while (angle < -pi)
    {
        angle = angle + 2*pi;
    }
    for (i = 0; i <= terme - 1; i = i + 1)
    {
        valeur = valeur + (z*((puissance(a,angle))/(factorielle(a))));
        a = a + 2;
        z = z * -1;
    }
    return valeur;
}
```

Figure 9: Code en langage C pour trouver le cosinus d'un angle.

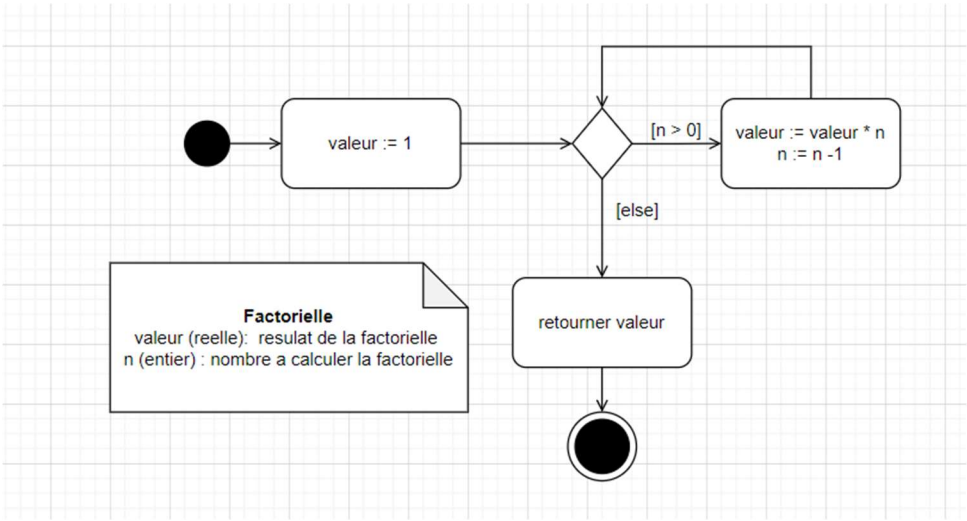

Figure 10: Diagramme de fonctions pour trouver la factorielle.

```c
float factorielle(unsigned int n)
{
    float valeur = 1;
    while (n > 0)
    {
        valeur = valeur * n;
        n = n - 1;
    }
    return valeur;
}
```

Figure 11: Code en langage C pour trouver la factorielle.

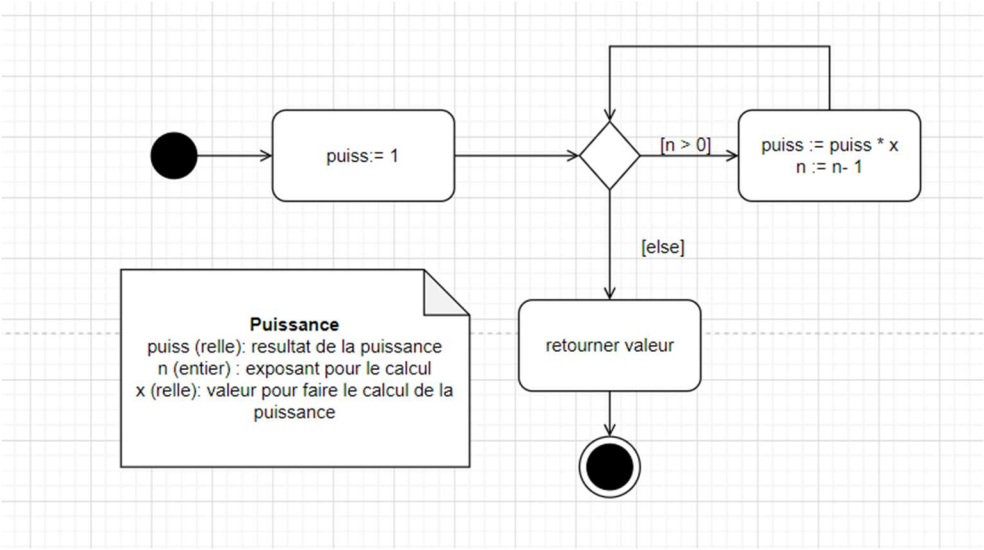

Figure 13: Diagramme de fonctions pour trouver la puissance.

```c
float puissance(unsigned int n, float x)
{
    float p = 1;
    while (n > 0)
    {
        p = p * x;
        n = n - 1;
    }
    return p;
}
```

Figure 14: Code en langage C pour trouver la puissance.

2.5 ADDITION DE DEUX MATRICES

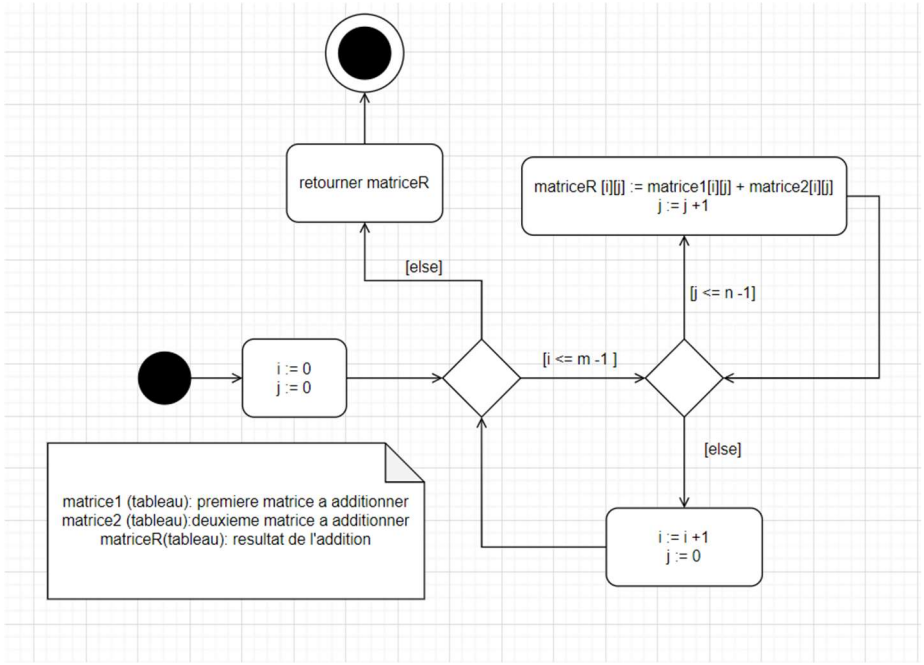

Figure 15: Diagramme de fonctions pour additionner deux matrices.

```c
void additionMatrice(int matrice1[m][n],int matrice2[m][n],int matriceR[m][n])
{
    for (int i = 0; i <= m-1; i++)
    {
        for (int j = 0; j <= n-1; j++)
        {
            matriceR[i][j] = matrice1[i][j] + matrice2[i][j];
        }
    }
}
```

Figure 16: Code en langage C pour additionner deux matrices.

2.6 MULTIPLICATION DE DEUX MATRICES

FONCTION multiplicationMatrice(matrice1(tableau), matrice2(tableau) matriceR (tableau))

//Trouve la multiplication entre deux matrices

//matrice1(tableau) : première matrice à multiplier

//matrice2(tableau) : deuxième matrice à multiplier

//matriceR(tableau) : résultat de la multiplication

DEBUT

// i (entier) : compteur incrémental

//j(entier) : compteur incrémental

//k(entier) : compteur incrémental

//n(entier) : nombre de ligne et colonne dans la matrice carrée

n := 2

POUR i = 0 **À** n – 1

POUR j = 0 **À** n -1

matriceR[i][j] := 0

POUR k = 0 **À** n – 1

matriceR[i][j] := matriceR[i][j] + matrice1[i][j]*matrice2[i][j]

FIN

```
void multiplicationMatrice(int matrice1[n][n],int matrice2[n][n],int matriceR[n][n])
{
  for(int i = 0; i <= n-1 ; i++)
  {
    for(int j = 0; j <= n-1; j++)
    {
      matriceR[i][j]=0;
      for(int k = 0; k < n; k++)
      {
        matriceR[i][j] = matriceR[i][j] + matrice1[i][k] * matrice2[k][j];
      }
    }
  }
}
```

Figure 17: Code en langage C pour multiplier deux matrices.

2.7 PLAN DE TEST

Tableau 1: Plan de test

Fonction	Paramètre	Attendue
Palindrome	kayakp	0
Palindrome	1234321	1
Palindrome	abcmmmmdef	0
Palindrome	Civic	0
Palindrome	lamarieeiramal	1
Recherche Caractère	1577 « 5 »	1
Recherche Caractère	Fitness « 3 »	-1
Recherche Caractère	chien « b »	-1
Recherche Caractère	Allo « A »	0
Sin	10	-0,5440
Sin	-4	0,7568
Sin	π/3	0.8660
Sin	Π	0
Cos	10	-0,8391
Cos	-4	-0,6536
Cos	π/3	0.5
Cos	π	-1

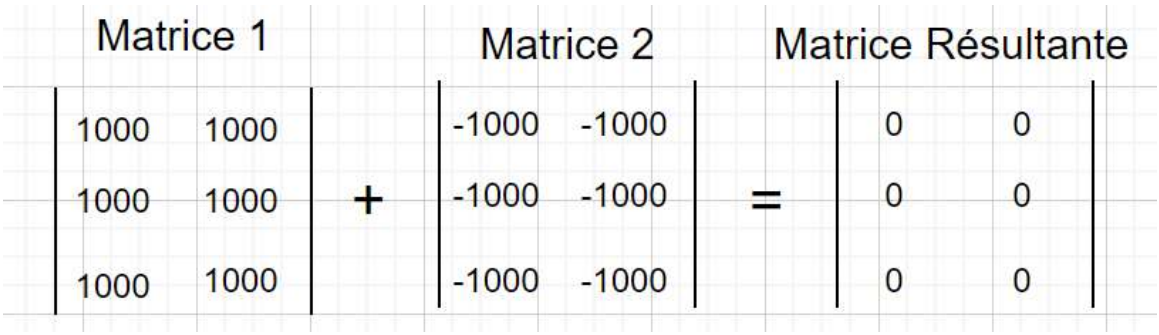

Figure 18 : Plan de test pour la fonction d'addition matrices

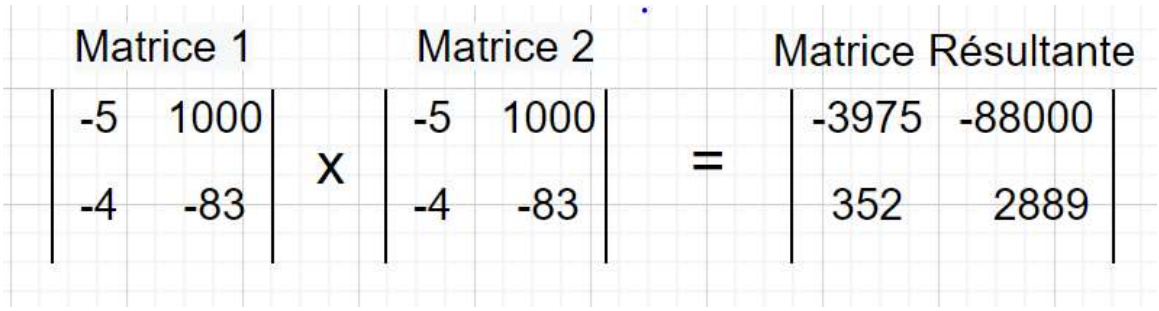

Figure 19 : Plan de test pour la fonction de multiplication de matrices

3. CONCLUSION

Pour conclure, cet APP permet d'acquérir de bonne habitude de programmation qui vont servir aux étudiants en génie électrique, génie informatique et génie robotique.

4. REFERENCES

[1] GUIDE DE L'ÉTUDIANTE ET DE L'ÉTUDIANT S1 – APP2sntetm (GEGIGRO), Automne 2022.