

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

RAPPORT APP 2

Introduction à la programmation et Programmation et algorithmes
GEN143, GEN 144

Présenté à
Charles-Antoine Brunet
Jean-Samuel Lauzon
Domingo Palao Munoz
Jonathan Vincent

Présenté par
Équipe numéro 15
Benjamin Gélinais – GELB2602
Simon Leroux – LERS0601

Sherbrooke – 6 octobre 2020

TABLE DES MATIÈRES

1.	Développement	4
1.1	Tableau des plans de test	4
1.2	Code, diagramme et pseudocode	6

LISTE DES FIGURES

Code 1	6
Code 2	7
Code 3	8
Code 4	9
Code 5	10
Code 6	11
Code 7	12
Code 8	12
UML 1	6
UML 2	7
UML 3	11

LISTE DES TABLEAUX

Tableau 1: Plan de tests des fonctions	4
--	---

1. DÉVELOPPEMENT

1.1 TABLEAU DES PLANS DE TEST

Tableau 1: Plan de tests des fonctions

Fonction	Paramètres	Résultat attendu
Recherche caractère	Mot : salut Caractère : s	Position : 0
Recherche caractère	Mot : Benjamin Caractère : b	Position : -1
Recherche caractère	Mot : simon Caractère : n	Position : 4
Recherche caractère	Mot : jardin Caractère : z	Position : -1
Palindrome	Mot : kayak	1
Palindrome	Mot : service	0
Palindrome	Mot : Laval	0
Palindrome	Mot : pavillon	0
Sinus	1	.8414
Sinus	0	0
Sinus	$2\pi/3$.8660
Sinus	$\pi/5$.5877
Cosinus	0	1
Cosinus	π	-1
Cosinus	$\pi/6$.8660
Cosinus	$\pi/8$.9238

Addition Matrice	A : $\{\{1,1,1\}, \{1,1,1\}\}$ B : $\{\{1,1,1\}, \{1,1,1\}\}$	$\{\{2,2,2\}, \{2,2,2\}\}$
Addition Matrice	A : $\{\{2,3,4\}, \{5,6,7\}\}$ B : $\{\{6,5,4\}, \{3,2,1\}\}$	$\{\{8,8,8\}, \{8,8,8\}\}$
Addition Matrice	A : $\{\{-1,-2,-3\}, \{-4,-5,-6\}\}$ B : $\{\{-6,-5,-4\}, \{-3,-2,-1\}\}$	$\{\{-7,-7,-7\}, \{-7,-7,-7\}\}$
Addition Matrice	A : $\{\{0,0,0\}, \{0,0,0\}\}$ B : $\{\{0,0,0\}, \{0,0,0\}\}$	$\{\{0,0,0\}, \{0,0,0\}\}$
Multiplication Matrice	A : $\{\{0,0,0\}, \{0,0,0\}, \{0,0,0\}\}$ B : $\{\{0,0,0\}, \{0,0,0\}, \{0,0,0\}\}$	$\{\{0,0,0\}, \{0,0,0\}, \{0,0,0\}\}$
Multiplication Matrice	A : $\{\{1,2,3\}, \{4,5,6\}, \{7,8,9\}\}$ B : $\{\{1,2,3\}, \{4,5,6\}, \{7,8,9\}\}$	$\{\{30,36,42\}, \{66,81,96\}, \{102,126,150\}\}$
Multiplication Matrice	A : $\{\{5,5,5\}, \{5, 5,5\}, \{5,5,5\}\}$ B : $\{\{5,5,5\}, \{5, 5,5\}, \{5,5,5\}\}$	$\{\{75,75,75\}, \{75, 75,75\}, \{75,75,75\}\}$
Multiplication Matrice	A : $\{\{-1,-2,-3\}, \{-4,-5,-6\}, \{-7,-8,-9\}\}$ B : $\{\{1,2,3\}, \{4,5,6\}, \{7,8,9\}\}$	$\{\{-30,-36,-42\}, \{-66,-81,-96\}, \{-102,-126,-150\}\}$

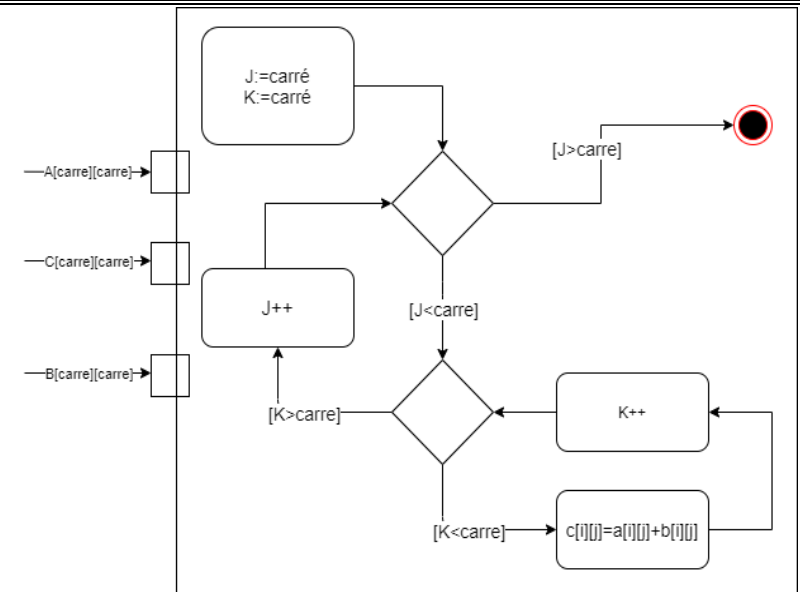
1.2 CODE, DIAGRAMME ET PSEUDOCODE

Code

```
void AdditionMat(int a[rangee][colonne],int b[rangee][colonne],int c[rangee][colonne]){  
    for(int i=0;i<rangee;i++){  
        for(int j=0;j<colonne;j++){  
            c[i][j]=a[i][j]+b[i][j];  
        }  
    }  
}
```

Code 1

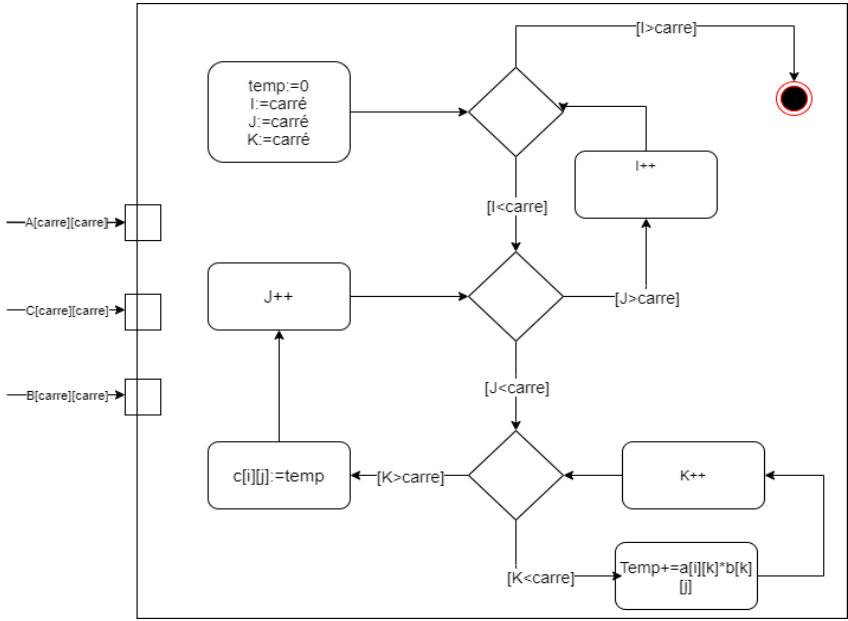
Diagramme/Pseudocode



UML 1

```
void MultiMat(int a[carre][carre],int b[carre][carre],int c[carre][carre]){
    int temp=0;
    for (int i=0; i < 3; i++){
        for (int j=0; j < 3; j++) {
            for (int k=0; k < 3; k++) {
                temp+=a[i][k]*b[k][j];
            }
            c[i][j] = temp;
            temp=0;
        }
    }
}
```

Code 2



UML 2

```
double cosinus(double angle)
{
    double somme = 1;
    int j = 2;

    for (int i = 2; i < termes; i++){
        if ((i%2)==0)
        {
            somme -= (pwr(angle,j))/(factoriel(j));
        }
        else
        {
            somme += (pwr(angle,j))/(factoriel(j));
        }
        j+=2;
    }

    return somme;
}
```

Code 3

FONCTION cosinus(x)

// x (réelle) : la valeur de l'angle en radian

DÉBUT

// somme (réelle) : sommation des termes dans la série pour approximer la valeur du

cosinus

//i (entier) : compteur 1

/j (entier) : compteur 2

somme := 1

j := 2

POUR i := 2 À termes PAR PAS DE 1

SI i%2==0

somme=somme- (exposant(x,j))/(factorielle(j))

SINON

somme=somme+(exposant(x,j))/(factorielle(j))

j = j+2

retourner somme

FIN


```

double sinus(float angle){
    double reponse=angle;
    int j=3;
    for(int i=2;i<iteration; i++){
        if((i%2)==0){
            reponse-= (pwr(angle,j))/(factoriel(j));
        }
        else{
            reponse+=(pwr(angle,j))/(factoriel(j));
        }
        j+=2;
    }
    return reponse;
}

```

Code 4

FONCTION sinus(angle):reponse

//angle(réel)

//reponse(réel)

DÉBUT

//j(entier)

//reponse:=angle

//j:=3

POUR i:=2 À iteration PAR PAS DE 1

SI i%2 est égal à 0 ALORS

reponse:= reponse - pwr(angle,j)/factoriel(j)

SINON

reponse:= reponse + pwr(angle,j)/factoriel(j)

j=j+2

Retourner la valeur de reponse

FIN

```

int rechcaractere(char caractere, char tableaucaractere[])
{
    int i = 0;
    int longueur_mot = 0;

    while (tableaucaractere[i] != '\0')
    {
        longueur_mot++;
        i++;
    }

    for (int j=0; j < longueur_mot; j++)
        if (tableaucaractere[j] == caractere)
        {
            return j;
        }

    return -1;
}

```

Code 5

FONCTION rechcaractere(caractere, tableaucaractere): position caractère

// caractere (caractère) : le caractère recherché

// tableaucaractere (tableau de caractères) : tableau de caractère séparé par des caractères

// longueur_mot (entier) : longueur du mot

// i (entier) : compteur #1

// j (entier) : compteur #2

DÉBUT

i:= 0

TANT QUE tableaucaractere[i] != '\0'

longueur_mot = longueur_mot + 1

i := i + 1

j:= 0

POUR j À longueur_mot PAR PAS DE 1

SI tableaucaractere[j] == caractere ALORS

retourner i

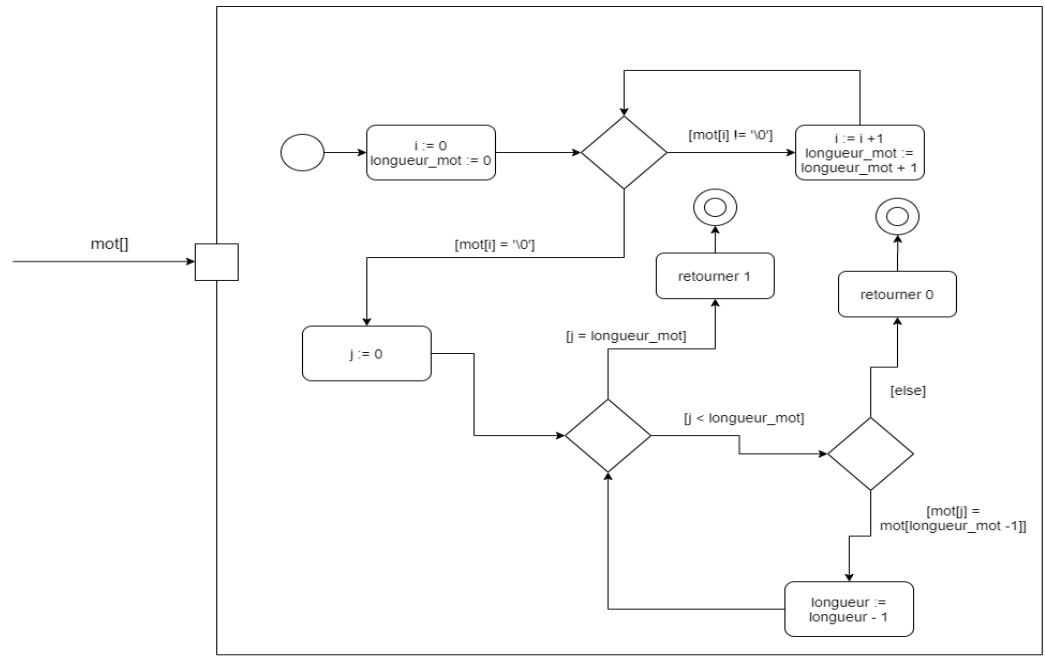
retourner -1

FIN

```
int palindrome(char mot[])
{
    int i = 0;
    int longueur_mot = 0;

    while (mot[i] != '\0')
    {
        longueur_mot++;
        i++;
    }

    for (int j=0; j <= longueur_mot; j++){
        if (mot[j] == mot[longueur_mot - 1])
        {
            longueur_mot--;
        }
        else
        {
            return 0;
        }
    }
    return 1;
}
```

Code 6

UML 3

```
float pwr(float base,int ala){
    float reponse=base;
    for(int i=1;i<ala; i++){
        reponse*=base;
    }
    return reponse;
}
```

Code 7

```
long double factoriel(int nombre){
    long double reponse = 1;
    for(int i=1; i<=nombre; i++){
        reponse*=i;
    }
    return reponse;
}
```

Code 8

FONCTION pwr(base,ala):reponse

//base(réel)

//ala(entier)

//reponse(réel)

DÉBUT

//reponse:=base

POUR i:=1 À i<ala PAR PAS DE 1

reponse:= reponse*base

Retourner la valeur de reponse

FIN

FONCTION factoriel(nombre):reponse

//nombre(entier)

//reponse(réel long)

DÉBUT

//reponse:=1

POUR i:=1 À i<=nombre PAR PAS DE 1

reponse:= reponse*i

retourner la valeur de reponse

FIN