

Tp3

Exercice 1

A screenshot of a code editor window titled "partie1.h — TP3". The file contains the following C++ code:

```
h partie1.h x C++ main.cpp
exercice1 > h partie1.h > Point > afficher()
1 class Point
2 {
3     protected:
4         int x, y;
5
6     public:
...
7     void afficher();
8     void initialiser(int a,int b);
9     void deplacer(int dx, int dy);
10    float distance(Point p1, Point p2);
11};

Ln 7, Col 21  Spaces: 4  UTF-8  CRLF  C++  Go Live  Mac  Prettier
```

A screenshot of a code editor window titled "partie1.cpp — TP3". The file contains the following C++ code:

```
h partie1.h C++ partie1.cpp 1 x C++ main.cpp
exercice1 > C++ partie1.cpp > initialiser(int, int)
3 #include<cmath>
4 using namespace std;
5
6 void Point::afficher(){
7     cout<<"l'abscisse du point est:"<<this->x<<endl;
8     cout<<"l'ordonnee du point est:"<<this->y<<endl;
9 }
10 Point::Point(int a, int b){
11     this->x=a;
12     this->y=b;
13 }
14 void Point::initialiser(int a,int b){
15     this->x=a;
16     this->y=b;
17 }
18 void Point::deplacer(int dx, int dy){
19     this->x=this->x+dx;
20     this->y=this->y+dy;
21 }
22 float Point::distance(Point p1, Point p2){
23     float d;
24     float e=pow(p2.x - p1.x,2)+pow(p2.y - p1.y,2);
25     d=sqrt(e);
26     return d;
27 }
28

Ln 14, Col 18  Spaces: 4  UTF-8  CRLF  C++  Go Live  Mac  Prettier
```

A status bar at the bottom right indicates "Do Not Disturb Mode is Enabled".

The screenshot shows a code editor window titled "main.cpp — TP3". The left sidebar contains icons for file operations like Open, Save, Find, Replace, and others. The main pane displays the following C++ code:

```
partie1.h    C++ main.cpp ×  
exercice1 > C++ main.cpp > ...  
1 #include <iostream>  
2 #include "partie1.h"  
3  
4 using namespace std;  
5  
6 int main()  
7 {  
8     Point pt1(3,5);  
9     pt1.afficher();  
10    pt1.deplacer(4,2);  
11    cout<<"apres deplacement:"<<endl;  
12    pt1.afficher();  
13    Point pt2(8,9);  
14    cout<<"affichage du deuxieme point"<<endl;  
15    pt2.afficher();  
16    float d;  
17    d=pt2.distance(pt1,pt2);  
18    cout<<"la distance entre les deux points est:"<<d<<endl<<endl<<endl;  
19    cout<<"partie cercle"<<endl;  
20
```

At the bottom of the editor, status indicators show "Ln 1, Col 1", "Tab Size: 4", "UTF-8", "CRLF", "C++", "Go Live", "Mac", "Prettier", and a few other icons.

- ayabenabbour@MacBookAirdeAYA exercice1 % g++ *.cpp
- ayabenabbour@MacBookAirdeAYA exercice1 % ./a.out

```
l'abscisse du point est:3
l'ordonnee du point est:5
apres deplacement:
l'abscisse du point est:7
l'ordonnee du point est:7
affichage du deuxieme point
l'abscisse du point est:8
l'ordonnee du point est:9
la distance entre les deux points est:2.23607
```

b)

partie1.h — TP3

```
partie1.h  x
exercice1 > h partie1.h > Cercle > point
1 class Point
2 {
3     protected:
4         int x, y;
...
6     public:
7         void afficher();
8         Point(int a, int b);
9         void deplacer(int dx, int dy);
10        float distance(Point p1, Point p2);
11    };

```

Ln 18, Col 27 Spaces: 4 UTF-8 CRLF C++ ⚡ Go Live Mac ⚡ Prettier ⚡ ⚡

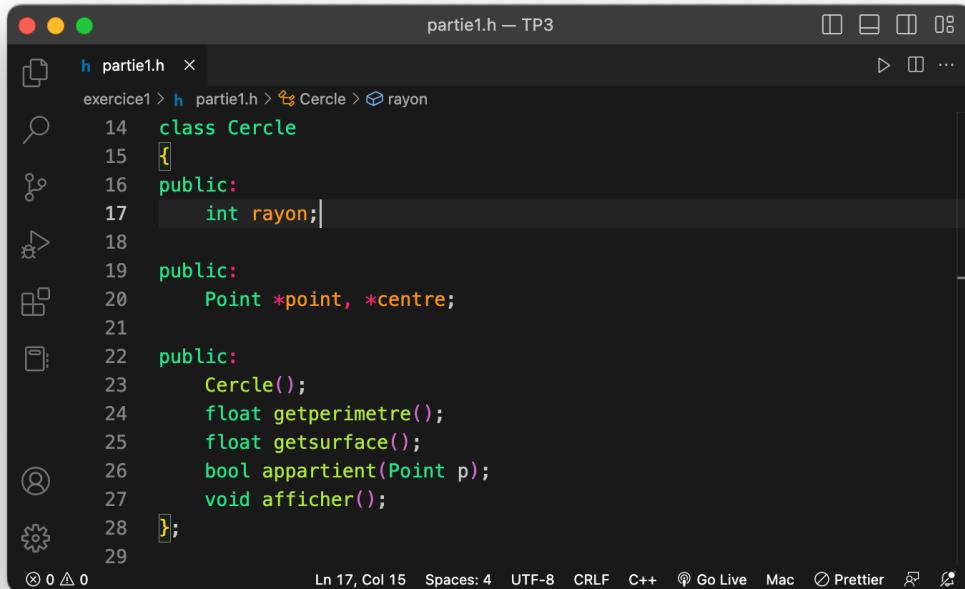
partie1.cpp — TP3

```
partie1.h  C++ partie1.cpp  x
exercice1 > C++ partie1.cpp > ...
1 #include "partie1.h"
2 #include<iostream>
3 #include<cmath>
4 using namespace std;
5
6 void Point::afficher(){
7     cout<<"l'abscisse du point est:"<<this->x<<endl;
8     cout<<"l'ordonnee du point est:"<<this->y<<endl;
9 }
10 Point::Point(int a, int b){
11     this->x=a;
12     this->y=b;
13 }
14 void Point::deplacer(int dx, int dy){
15     this->x=this->x+dx;
16     this->y=this->y+dy;
17 }
18 float Point::distance(Point p1, Point p2){
19     float d;
20     float e=pow(p2.x - p1.x,2)+pow(p2.y - p1.y,2);
21     d=sqrt(e);
22     return d;
23 }
24
25
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF C++ ⚡ Go Live Mac ⚡ Prettier ⚡ ⚡

```
l'abscisse du point est:3
l'ordonnee du point est:5
apres deplacement:
l'abscisse du point est:7
l'ordonnee du point est:7
affichage du deuxieme point
l'abscisse du point est:8
l'ordonnee du point est:9
la distance entre les deux points est:2.23607
```

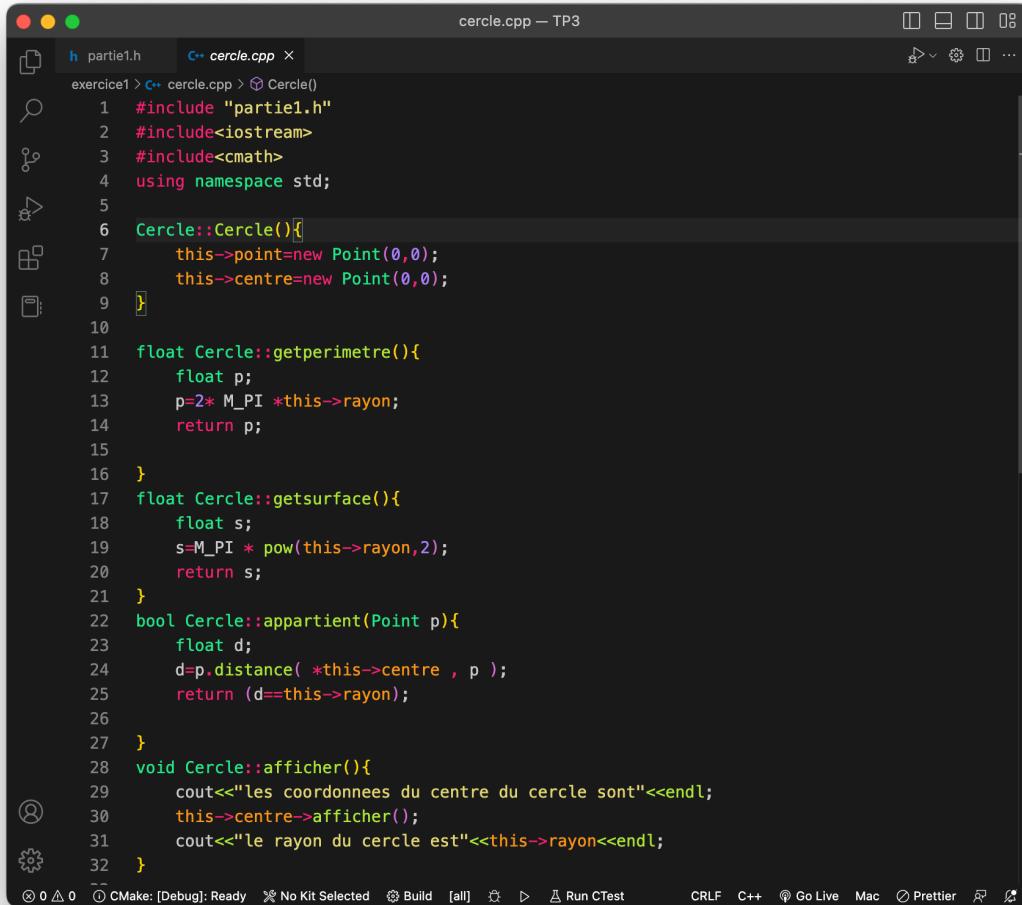
Partie 2:



partie1.h — TP3

```
h partie1.h
exercice1 > h partie1.h > Cercle > rayon
14 class Cercle
15 {
16 public:
17     int rayon;
18
19 public:
20     Point *point, *centre;
21
22 public:
23     Cercle();
24     float getperimetre();
25     float getsurface();
26     bool appartient(Point p);
27     void afficher();
28 }
```

Ln 17, Col 15 Spaces: 4 UTF-8 CRLF C++ ⚡ Go Live Mac ⚡ Prettier



cercle.cpp — TP3

```
h partie1.h C cercle.cpp
exercice1 > C cercle.cpp > Cercle()
1 #include "partie1.h"
2 #include<iostream>
3 #include<cmath>
4 using namespace std;
5
6 Cercle::Cercle(){
7     this->point=new Point(0,0);
8     this->centre=new Point(0,0);
9 }
10
11 float Cercle::getperimetre(){
12     float p;
13     p=2* M_PI *this->rayon;
14     return p;
15 }
16
17 float Cercle::getsurface(){
18     float s;
19     s=M_PI * pow(this->rayon,2);
20     return s;
21 }
22
23 bool Cercle::appartient(Point p){
24     float d;
25     d=p.distance( *this->centre , p );
26     return (d==this->rayon);
27 }
28
29 void Cercle::afficher(){
30     cout<<"les coordonnees du centre du cercle sont"<<endl;
31     this->centre->afficher();
32     cout<<"le rayon du cercle est"<<this->rayon<<endl;
33 }
```

CMake: [Debug] Ready No Kit Selected Build [all] ⚡ ⚡ Run CTest CRLF C++ ⚡ Go Live Mac ⚡ Prettier ⚡

The screenshot shows a dark-themed code editor window titled "main.cpp — TP3". The code is written in C++ and defines two circles, cercle1 and cercle2, with their centers and radii. It then checks if point pt(120, 100) belongs to both circles and prints the results.

```
partie1.h      C++ main.cpp ×
exercice1 > C++ main.cpp > main()
1 #include <iostream>
2 #include "partie1.h"
3
4 using namespace std;
5
6 int main()
7 {
8     Point centre(100,100);
9     Point point(200,200);
10    Cercler cercle1;
11    cercle1.centre=&centre;
12    cercle1.point=&point;
13    float rayon1=centre.distance(centre,point);
14    cercle1.rayon=rayon1;
15
16    Cercler cercle2;
17    Point centre2(130,100);
18    cercle2.centre=&centre2;
19    cercle2.rayon=40;
20
21    cercle1.afficher();
22    cercle2.afficher();
23
24    float p=cercle1.getperimetre();
25    float p2=cercle2.getperimetre();
26    cout<<"le rayon du premier cercle est"<<cercle1.rayon<<endl;
27    cout<<"le rayon du deuxième cercle est"<<cercle2.rayon<<endl;
28    Point pt(120,100);
29    bool a=cercle1.appartient(pt);
30    bool b=cercle2.appartient(pt);
31    if (a==true && b==true){
32
33        cout <<"le point";
34        pt.afficher() ;
35        cout<<"appartient à l'intersection des deux cercles"<<endl;
36
37
38    return 0;
39 }
```

At the bottom of the editor, there are tabs for "CMake: [Debug]: Ready", "No Kit Selected", "Build [all]", "Run CTest", and various tool icons. The status bar shows "Tab Size: 4", "UTF-8", "CRLF", "C++", "Go Live", "Mac", "Prettier", and file paths.

```
partie cerce
les coordonnees du centre du cerce sont
l'abscisse du point est:100
l'ordonnee du point est:100
le rayon du cerce est141
les coordonnees du centre du cerce sont
l'abscisse du point est:130
l'ordonnee du point est:100
le rayon du cerce est40
le rayon du premier cerce est141
le rayon du deuxième cerce est40
ayabenkabbour@MacBookAirdeAYA exercice1 %
```

Exercice2

The screenshot shows the Xcode IDE interface with the file `exercice2complet.cpp` open. The code defines a class `Vecteur` with private members `x`, `y`, and `hauteur`. It includes a constructor, a destructor, and two static methods for vector operations: `prod_scal` and `somme`.

```
1 #include <iostream>
2 #include<cmath>
3 using namespace std;
4 //partie fichier.h
5 class Vecteur{
6     public:
7         float x;
8         float y;
9         float hauteur;
10    public:
11        void affiche();
12        Vecteur();
13        Vecteur(float x,float y, int c=0);
14        ~Vecteur();
15        float prod_scal(Vecteur* vecteur1, Vecteur* vecteur2);
16        Vecteur somme(Vecteur* vecteur1, Vecteur* vecteur2);
17
18
```

At the bottom, there are status icons for CMake, Debug, No Kit Selected, Build, Run CTest, CRLF, C++, Go Live, Mac, Prettier, and a few others.

The screenshot shows the Xcode IDE interface with the file `exercice2complet.cpp` open. The code implements the methods defined in `Vecteur.h`. It includes implementations for `affiche`, `Vecteur()`, `Vecteur(float x, float y, int c)`, `~Vecteur()`, `prod_scal`, and `somme`.

```
19 ;
20 //partie fichier.cpp (implementation des fichiers)
21 void Vecteur::affiche(){
22     cout<<"l'abscisse du vecteur est:" << this->x<<"--"<<endl;
23     cout<<"l'ordonnee du vecteur est:" <<this->y<<"--"<<endl;
24     cout<<"la hauteur du vecteur est:" <<this->hauteur<<"--"<<endl;
25 }
26 Vecteur::Vecteur(){
27     this->x=0;
28     this->y=0;
29     this->hauteur=0;
30 }
31 Vecteur::Vecteur(float x,float y, int c){
32     this->x=x;
33     this->y=y;
34     this->hauteur=c;
35 }
36 Vecteur::~Vecteur(){
37     cout<<"le vecteur est detruit son adresse est:"<<this<<endl;
38 }
39 float Vecteur::prod_scal(Vecteur* vecteur1, Vecteur* vecteur2){
40     float p=(vecteur2->x * vecteur1->x)+(vecteur2->y * vecteur1->y);
41     return p;
42 }
43 Vecteur Vecteur::somme(Vecteur* vecteur1, Vecteur* vecteur2){
44     Vecteur somme;
45     somme.x=vecteur2->x + vecteur1->x;
46     somme.y=vecteur2->y + vecteur1->y;
47     return somme;
48 }
```

At the bottom, there are status icons for CMake, Debug, No Kit Selected, Build, Run CTest, CRLF, C++, Go Live, Mac, Prettier, and a few others.

The screenshot shows a macOS terminal window with a dark theme. At the top, there are three tabs: "partie1.h", "C++ main.cpp", and "C++ exercice2complet.cpp x". The "exercice2complet.cpp" tab is active. The code editor displays the following C++ code:

```
partie1.h          C++ main.cpp          C++ exercice2complet.cpp x
exercice2 > C++ exercice2complet.cpp > prod_scal(Vecteur *, Vecteur *)
51 //partie fichier main.cpp (fonction main)
52 int main() {
53     Vecteur vecteur;
54     cout<<"affichage des coordonnees du vecteur 1 initialiser par constructeur par defaut"<<endl;
55     vecteur.affiche();
56     cout<<endl;
57     cout<<"affichage des coordonnees du vecteur 2"<<endl;
58     Vecteur vecteur1(12,4);
59     vecteur1.affiche();
60     cout<<endl;
61     Vecteur vecteur2(23,8);
62     cout<<"affichage du vecteur 3"<<endl;
63     vecteur2.affiche();
64     cout<<endl;
65     float produit=vecteur1.prod_scal(&vecteur1 , &vecteur2);
66     cout<<"le produit scalaire des deux vecteurs est egale a:"<<produit<<endl;
67
68     Vecteur somme=vecteur1.somme(&vecteur1 , &vecteur2);
69     cout<<"le vecteur somme des deux produit est ";
70     somme.affiche();
71     cout<<endl;
72 }
73
```

At the bottom of the terminal window, there are several status indicators: "0 0 △ 0", "CMake: [Debug]: Ready", "No Kit Selected", "Build [all]", "Run CTest", "UTF-8 CRLF C++ Go Live Mac Prettier".

```
● ayabenkabbour@MacBookAirdeAYA TP3 % cd "/Users/ayabenkabbour/Desktop/cpp/TP3/exercice2/" && g++ exercice2complet.cpp -o exercice2complet && "/Users/ayabenkabbour/Desktop/cpp/TP3/exercice2/"exercice2complet
affichage des coordonnees du vecteur 1 initialiser par constructeur par defaut
l'abscisse du vecteur est:0--
l'ordonnee du vecteur est:0--
la hauteur du vecteur est:0--

affichage des coordonnees du vecteur 2
l'abscisse du vecteur est:12--
l'ordonnee du vecteur est:4--
la hauteur du vecteur est:0--

affichage du vecteur 3
l'abscisse du vecteur est:23--
l'ordonnee du vecteur est:8--
la hauteur du vecteur est:0--

le produit scalaire des deux vecteurs est egale a:308
le vecteur somme des deux produit est l'abscisse du vecteur est:35--
l'ordonnee du vecteur est:12--
la hauteur du vecteur est:0--

le vecteur est detruit son adresse est:0x16d1234ec
le vecteur est detruit son adresse est:0x16d1234fc
le vecteur est detruit son adresse est:0x16d123508
le vecteur est detruit son adresse est:0x16d123524
○ ayabenkabbour@MacBookAirdeAYA exercice2 %
```

Exercice 3

The screenshot shows a dark-themed code editor window titled "exercice3complet.cpp — TP3". The code defines a class `Pile` with protected members `taille`, `*pile`, `taille_max`, and `courant`. It includes a constructor `Pile(int n)`, a default constructor `Pile()`, a destructor `~Pile()`, and four public member functions: `empiler(int p)`, `depiler()`, `pleine()`, and `vide()`.

```
#include <iostream>
#include <new>

using namespace std;

//fichier.h
class Pile{
protected:
    int taille;
    int *pile;
    int taille_max;
    int courant;
public:
    Pile(int n);
    Pile();
    ~Pile();
    void empiler(int p);
    int depiler();
    int pleine();
    int vide();
};

CMake: [Debug]: Ready  No Kit Selected  Build [all]  Run CTest  Mac  Prettier
```

The screenshot shows the same code editor window, but the code has been scrolled down to show the implementation of the constructor `Pile::Pile()` and the constructor `Pile::Pile(int n)`. The constructor `Pile::Pile()` allocates memory for the stack and initializes it. The constructor `Pile::Pile(int n)` does the same but initializes the stack size to `n`.

```
//fichier.cpp
Pile::Pile(){
    this->pile=new int[20];
    if(pile==NULL)
        cout<<" Echec d'allouement de memoire";
    else
    {
        taille_max=20;
        courant=0;
        taille = 0;
        cout<<" pile cree ";
    }
}
Pile::Pile(int n){
    this->pile=new int[n];
    if(pile==NULL)
        cout<<" Echec d'allouement de memoire";
    else
    {
        taille_max=n;
        courant=0;
        taille = 0;
        cout<<" pile cree";
    }
}

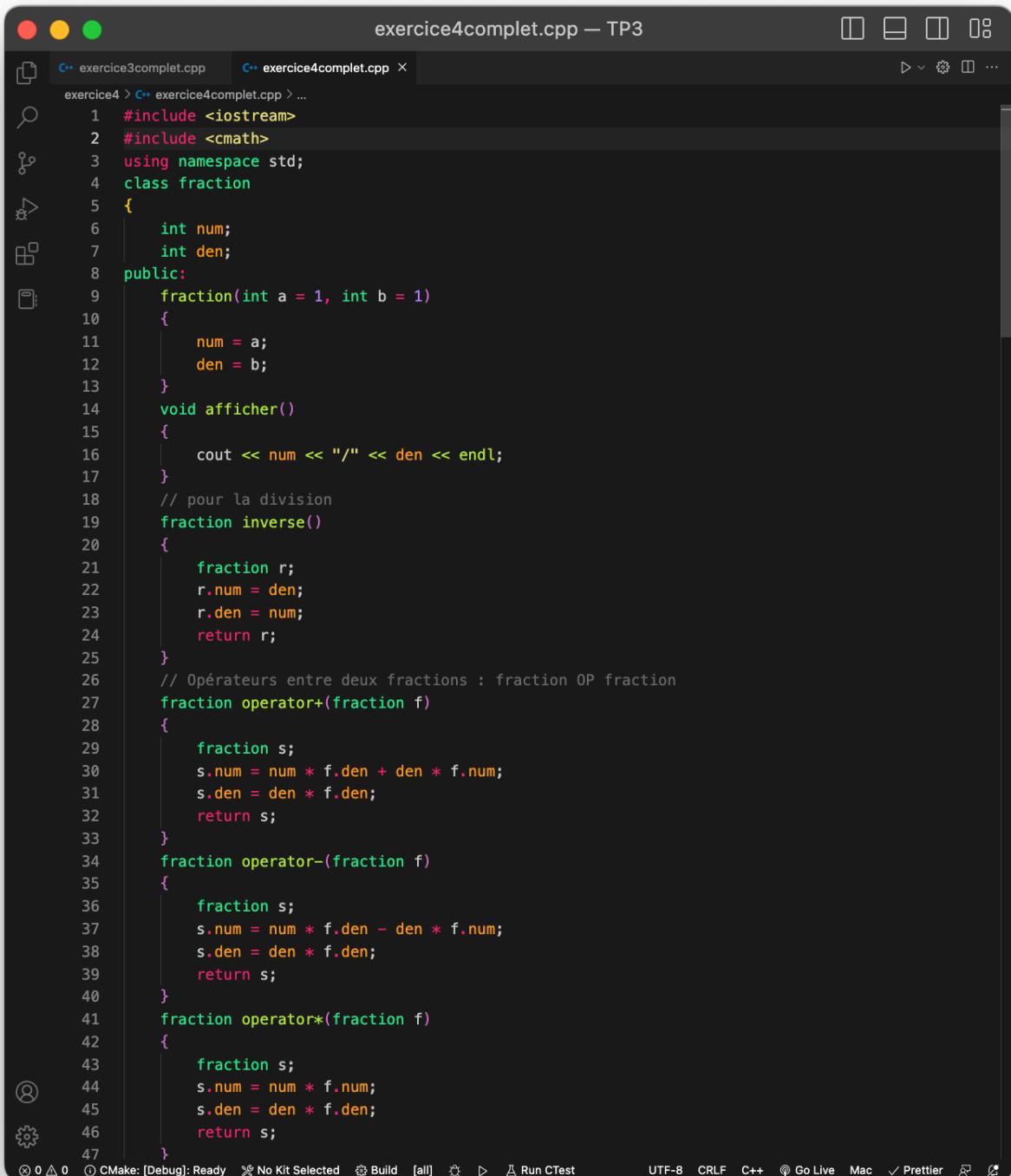
CMake: [Debug]: Ready  No Kit Selected  Build [all]  Run CTest  Mac  Prettier
```

```
exercice3complet.cpp — TP3
47 }
48 Pile::~Pile(){
49     cout<<"la pile est liberee";
50 }
51 void Pile::empiler(int p){
52     if(this->taille>this->taille_max){
53         int n=this->taille;
54         this->courant=pile[n];
55         this->pile[n]=p;
56         this->taille++; }
57     else{
58         cout<<"la pile a atteint sa taille maximale";
59     }
60 }
61 int Pile::depiler(){
62
63     int n=this->taille;
64     int s=this->pile[n];
65
66     this->taille--;
67     courant = pile[taille];
68     return s;
69 }
70 int Pile::pleine(){
71     return(this->taille==this->taille_max)? 1:0;
72 }
73 int Pile::vide(){
74     return(this->taille==0)? 1:0;
75 }
76 }
77 //fichier main.cpp
```

```
exercice3complet.cpp — TP3
77
78 //fichier main.cpp
79
80 int main() {
81     Pile pile(3);
82     cout<<endl;
83     pile.empiler(1);
84     pile.empiler(3);
85     pile.empiler(5);
86
87     int p=pile.pleine();
88     if(p==1)
89         cout<<"pile pleine"<<endl;
90     else
91         cout<<"pile non pleine";
92     int s=pile.depiler();
93     cout<<"l'element depile est"<<s<<endl;
94     s=pile.depiler();
95     cout<<"l'element depile est"<<s<<endl;
96     s=pile.depiler();
97     cout<<"l'element depile est"<<s<<endl;
98     int v=pile.vide();
99     if(v==1)
100        cout<<"pile vide"<<endl;
101    else
102        cout<<"pile non vide";
103
104
105    return 0;
106 }
```

```
● ayabenkabbour@MacBookAirdeAYA exercice3 % cd "/Users/ayabenkabbour/Desktop/cpp/TP3/exercice3/" && g++ exercice3complet.cpp -o exercice3complet && "/Users/ayabenkabbour/Desktop/cpp/TP3/exercice3/"exercice3complet
pile cree
pile pleine
l'element depile est0
l'element depile est5
l'element depile est3
pile vide
la pile est liberee%
```

Exercice 4



The screenshot shows the Xcode IDE interface with the file `exercice4complet.cpp` open. The code defines a class `fraction` with methods for construction, display, inverse, and arithmetic operators (+, -, *).

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4 class fraction
5 {
6     int num;
7     int den;
8 public:
9     fraction(int a = 1, int b = 1)
10    {
11        num = a;
12        den = b;
13    }
14    void afficher()
15    {
16        cout << num << "/" << den << endl;
17    }
18    // pour la division
19    fraction inverse()
20    {
21        fraction r;
22        r.num = den;
23        r.den = num;
24        return r;
25    }
26    // Opérateurs entre deux fractions : fraction OP fraction
27    fraction operator+(fraction f)
28    {
29        fraction s;
30        s.num = num * f.den + den * f.num;
31        s.den = den * f.den;
32        return s;
33    }
34    fraction operator-(fraction f)
35    {
36        fraction s;
37        s.num = num * f.den - den * f.num;
38        s.den = den * f.den;
39        return s;
40    }
41    fraction operator*(fraction f)
42    {
43        fraction s;
44        s.num = num * f.num;
45        s.den = den * f.den;
46        return s;
47    }
```

At the bottom of the Xcode window, there are several status icons and labels: CMake: [Debug]: Ready, No Kit Selected, Build [all], Run CTest, UTF-8 CRLF C++ Go Live Mac, Prettier.

exercice4complet.cpp — TP3

```
exercice3complet.cpp  exercice4complet.cpp x
exercice4 > C++ exercice4complet.cpp > fraction > operator-(fraction)
47     }
48     fraction operator/(fraction f)
49     {
50         return *this * (f.inverse());
51     }
52
53 // Opérateurs entre une fraction et un nombre
54 fraction operator+(int x)
55 {
56     fraction r(x, 1);
57     return *this + r;
58 }
59 fraction operator-(int x)
60 {
61     fraction r(x, 1);
62     return *this - r;
63 }
64 fraction operator*(int x)
65 {
66     fraction r(x, 1);
67     return *this * r;
68 }
69 fraction operator/(int x)
70 {
71     fraction r(x, 1);
72     return *this / r;
73 }
74
75 // Opérateurs entre un nombre et une fraction
76 friend fraction operator+(int, fraction);
77 friend fraction operator-(int, fraction);
78 friend fraction operator*(int, fraction);
79 friend fraction operator/(int, fraction);
80
81 // Opérateurs de comparaison
82 bool operator>(fraction f)
83 {
84     return (num * f.den > den * f.num);
85 }
86 bool operator>=(fraction f)
87 {
88     return (num * f.den >= den * f.num);
89 }
90 bool operator==(fraction f)
91 {
92     return (num * f.den == den * f.num);
93 }
```

① 0 △ 0 ② CMake: [Debug]: Ready ③ No Kit Selected ④ Build [all] ⑤ ⑥ ⑦ ⑧ Run CTest ⑨ UTF-8 CRLF C++ ⑩ Go Live Mac ⑪ Prettier ⑫ ⑬

exercice4complet.cpp — TP3

```
exercise4 > C++ exercice4complet.cpp > fraction > % operator-(fraction)
93     }
94     bool operator<(fraction f)
95     {
96         return !(*this >= f);
97     }
98     bool operator<=(fraction f)
99     {
100        return !(*this > f);
101    }
102    bool operator!=(fraction f)
103    {
104        return !(*this == f);
105    }
106 };
107 fraction operator+(int x, fraction f)
108 {
109     // On appelle l'opérateur : fraction + nombre
110     return f + x;
111 }
112 fraction operator-(int x, fraction f)
113 {
114     return f - x;
115 }
116 fraction operator*(int x, fraction f)
117 {
118     return f * x;
119 }
120 fraction operator/(int x, fraction f)
121 {
122     return f / x;
123 }
124 int main()
125 {
126     fraction f1(9, 5), f2(1, 2);
127     cout << "Fraction F1 : ";
128     f1.afficher();
129     cout << "Fraction F2 : ";
130     f2.afficher();
131
132     cout << "F1 + F2 = ";
133     (f1 + f2).afficher();
134
135     cout << "F1 - F2 = ";
136     (f1 - f2).afficher();
137
138     cout << "F1 * F2 = ";
139     (f1 * f2).afficher();
140 }
```

① ② ③ ④ ⑤ CMake: [Debug]: Ready ⑥ No Kit Selected ⑦ Build [all] ⑧ ⑨ ⑩ Run CTest ⑪ UTF-8 ⑫ CRLF ⑬ C++ ⑭ Go Live ⑮ Mac ⑯ Prettier ⑰ ⑱

exercice4complet.cpp — TP3

```
exercise4 > C++ exercice4complet.cpp > main()
138     cout << "F1 * F2 = ";
139     (f1 * f2).afficher();
140
141     cout << "F1 / F2 = ";
142     (f1 / f2).afficher();
143
144     cout << "F1 + 5 = ";
145     (f1 + 5).afficher();
146
147     cout << "3 * F2 = ";
148     (3 * f2).afficher();
149
150     if (f1 == f2)
151         cout << "Les deux fractions sont égales" << endl;
152     else
153         cout << "Les deux fractions ne sont pas égales" << endl;
154
155     if (f1 > f2)
156     {
157         cout << "La fraction la plus grande est : ";
158         f1.afficher();
159     }
160     else if (f1 < f2)
161     {
162         cout << "La fraction la plus grande est : ";
163         f2.afficher();
164     }
165 }
```

① ② ③ ④ ⑤ CMake: [Debug]: Ready ⑥ No Kit Selected ⑦ Build [all] ⑧ ⑨ ⑩ Run CTest ⑪ UTF-8 ⑫ CRLF ⑬ C++ ⑭ Go Live ⑮ Mac ⑯ Prettier ⑰ ⑱

Do Not Disturb Mode is Enabled

```
cc /complete.cpp -o exercice /complete dd /usr/bin/gj
Fraction F1 : 9/5
Fraction F2 : 1/2
F1 + F2 = 23/10
F1 - F2 = 13/10
F1 * F2 = 9/10
F1 / F2 = 18/5
F1 + 5 = 34/5
3 * F2 = 3/2
Les deux fractions ne sont pas égales
La fraction la plus grande est : 9/5
ayabenkabbour@MacBookAirdeAYA exercice4 %
```