

Plateforme de Développement Continu

Comprendre la Théorie pour mieux Pratiquer
Sciences de l'Ingénieur
Cours - Tutoriels

C++

Apprendre le C++

J'aime, Je partage
Montez en Compétences

Auteur

Je suis **Gérard KESSE**,
Ingénieur en Développement Informatique C/C++/Qt,
Avec à la fois des compétences en Systèmes Embarqués et en Robotique.

Formé à Polytech'Montpellier, Je suis un professionnel de conception de projets logiciel applicatif ou embarqué dans les secteurs de l'Aéronautique, de la Robotique, des Drones et de la Vision par Ordinateur. Aussi, Je reste ouvert à d'autres types de secteurs tels que l'Énergie et les Finances.

Les Sciences de l'Ingénieur sont au cœur du métier d'ingénieur. Sur le site **ReadyDev**, la Plateforme de Développement Continu, dont j'en suis le concepteur, vous trouverez des cours et des tutoriels adaptés aux sciences de l'ingénieur.

J'aime, Je partage.

Gérard KESSE

[GitHub](#) | [LinkedIn](#) | [SiteWeb](#)



Sommaire

Auteur.....	2
Sommaire	3
Introduction	5
Installation sous Windows avec MinGW	5
Téléchargements	5
Installation de MinGW	5
Installation de Notepad++	5
Compiler un projet C++ avec MinGW	6
Compiler un projet C++ avec un seul fichier source	6
Compiler un projet C++ avec plusieurs fichiers sources	9
Communiquer avec la console	12
Afficher un message	12
Communiquer avec le clavier	13
Lire des données à partir du clavier	13
Gérer le problème d'espace lors de la saisie de données	15
Gérer les conditions	17
Condition If (Si)	17
Condition Else (Sinon)	19
Condition Else If (Sinon Si)	21
Gérer les boucles	23
Boucle For (Pour Tout)	23
Boucle While (Tant Que)	25
Boucle Do While (Faire Tant Que)	27

Introduction

Le C++ est un langage de programmation orienté objet. Le but de ce tutoriel est de vous apprendre le C++.

Prérequis :

Aucun prérequis n'est nécessaire.

Installation sous Windows avec MinGW

Téléchargements

Notepad++ :

<https://notepad-plus-plus.org/fr/>

MinGW :

<http://www.mingw.org/>

Installation de MinGW

Packages MinGW :

mingw33-base

mingw32-gcc-g++

Installation de Notepad++

Plugins Notepad++ :

TextFX

NppExport

Compiler un projet C++ avec MinGW

Compiler un projet C++ avec un seul fichier source

Objectif :

Compiler un projet C++ avec un seul fichier source.

Créer un fichier source en C++.

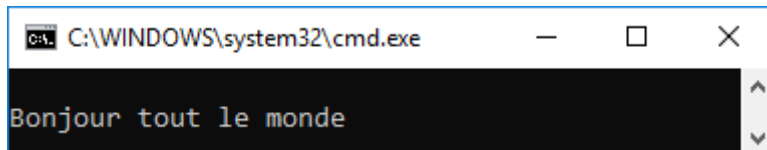
Créer un fichier de construction Makefile.

Implémentation :

Créer un fichier source (main.cpp).

Afficher un message dans la console (Bonjour tout le monde).

Résultat :



```
C:\WINDOWS\system32\cmd.exe
Bonjour tout le monde
```

Dossier projet :

```
src/main.cpp
win/cmd_build.bat
win/cmd_clean.bat
win/cmd_compile.bat
win/cmd_run.bat
win/bin/
win/build/
```

src/main.cpp

```
//=====
#include <iostream>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    cout << "Bonjour tout le monde\n";
    return 0;
}
//=====
```

win/Makefile

```
GSRC = ../src
GBIN = bin
GBUILD = build
GTARGET = $(GBIN)/GProject.exe

GOBJS = \
    $(GBUILD)/main.o

all: $(GOBJS)
    g++ -o $(GTARGET) $(GOBJS)
$(GBUILD)/main.o: $(GSRC)/main.cpp
    g++ -c $(GSRC)/main.cpp -o $(GBUILD)/main.o
clean:
    del /q $(GBIN)\* $(GBUILD)\*
```

Macros Makefile :

GSRC : répertoire des fichiers sources
GBIN : répertoire de génération du fichier exécutable
GBUILD : répertoire de génération des fichiers objets
GTARGET : chemin du fichier exécutable
GINCS : liste des répertoires de fichiers entêtes
GLIBS: liste des répertoires et des fichiers librairies.
GOBJS : liste des fichiers objets

win/cmd_build.bat

```
@echo off

call cmd_clean.bat
call cmd_compile.bat
call cmd_run.bat
```

win/cmd_clean.bat

```
@echo off

set PATH=C:\MinGW\bin

mingw32-make clean
```

win/cmd_compile.bat

```
@echo off
```

```
set PATH=C:\MinGW\bin
```

```
mingw32-make
```

win/cmd_run.bat

```
@echo off
```

```
set PATH=C:\MinGW\bin
```

```
echo.
```

```
bin\GProject.exe
```

```
echo.
```

```
pause
```

Nettoyage du projet :

```
cmd_clean.bat
```

Compilation du projet :

```
cmd_compile.bat
```

Exécution du projet :

```
cmd_run.bat
```

Construction du projet :

```
cmd_build.bat
```

Construction manuel :

```
del /q bin\* build\*
```

```
g++ -c ../src/main.cpp -o build/main.o
```

```
g++ -o bin/GProject.exe build/main.o
```


Compiler un projet C++ avec plusieurs fichiers sources

Objectif :

Compiler un projet C++ avec plusieurs fichiers sources.

Créer un fichier entête en C++.

Créer une fonction en C++.

Créer un fichier de construction Makefile.

Implémentation :

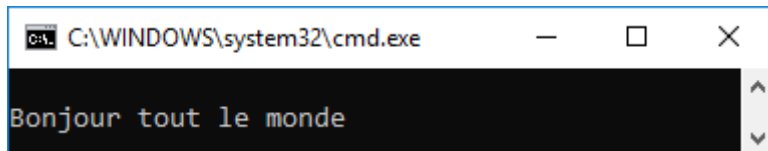
Créer un fichier entête (hello.h). Déclarer la fonction dire bonjour (sayHello()).

Créer un fichier source (hello.cpp). Définir la fonction dire bonjour (sayHello()).

Créer un fichier source (main.cpp). Utiliser la fonction dire bonjour (sayHello())

Afficher un message dans la console (Bonjour tout le monde).

Résultat :



Dossier projet :

src/main.cpp

src/manager/hello.h

src/manager/hello.cpp

win/Makefile

win/bin/

win/build/

src/main.cpp

```
//=====
#include "hello.h"
//=====
int main(int argc, char** argv) {
    sayHello();
    return 0;
}
//=====
```

src/manager/hello.h

```
//=====
#ifndef _hello_
#define _hello_
//=====
#include <iostream>
//=====
using namespace std;
//=====
void sayHello();
//=====
#endif
//=====
```

src/manager/hello.cpp

```
//=====
#include "hello.h"
//=====
void sayHello() {
    cout << "Bonjour tout le monde\n";
}
//=====
```

win/Makefile

```
GSRC = ../src
GBIN = bin
GBUILD = build
GTARGET = $(GBIN)/GProject.exe

GINCS = \
    -I../src/manager

GOBJS = \
    $(GBUILD)/main.o \
    $(GBUILD)/hello.o

all: $(GOBJS)
    g++ -o $(GTARGET) $(GOBJS)
$(GBUILD)/main.o: $(GSRC)/main.cpp
    g++ -c $(GSRC)/main.cpp -o $(GBUILD)/main.o
$(GINCS)
$(GBUILD)/hello.o: $(GSRC)/manager/hello.cpp
    g++ -c $(GSRC)/manager/hello.cpp -o
$(GBUILD)/hello.o $(GINCS)
clean:
    del /q $(GBIN)\* $(GBUILD)\*
```

Construction du projet :

Cmd_build.bat

Construction manuelle :

```
del /q bin\* build\*
g++ -c ../src/main.cpp -o build/main.o -
I../src/manager
g++ -c ../src/manager/hello.cpp -o build/hello.o -
I../src/manager
g++ -o bin/GProject.exe build/main.o build/hello.o
```

Communiquer avec la console

Afficher un message

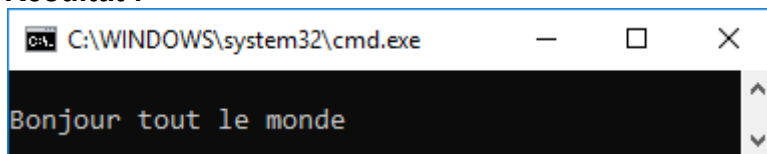
Objectif :

Afficher un message dans la console.

Implémentation :

Afficher un message dans la console (Bonjour tout le monde).

Résultat :



Dossier projet :

src/main.cpp
win/bin/
win/build/

src/main.cpp

```
//=====
#include <iostream>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    cout << "Bonjour tout le monde\n";
    return 0;
}
//=====
```

Communiquer avec le clavier

Lire des données à partir du clavier

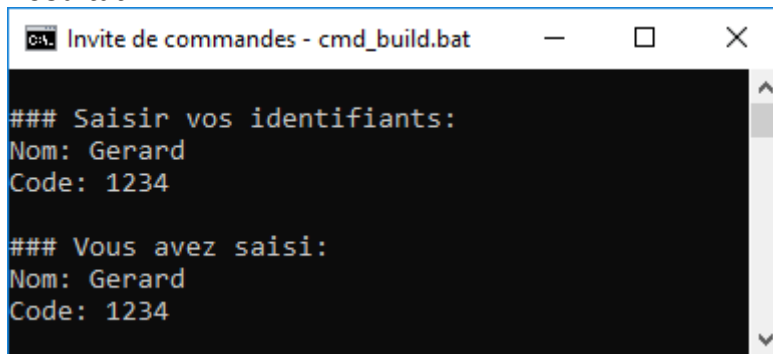
Objectif :

Récupérer des données saisies à partir du clavier.

Implémentation :

Récupérer les identifiants d'une personne saisis à partir du clavier.

Récupérer le nom (m_name) et le code d'entrée (m_code) de la personne.

Résultat :

```
### Saisir vos identifiants:
Nom: Gerard
Code: 1234

### Vous avez saisi:
Nom: Gerard
Code: 1234
```

Dossier projet :

src/main.cpp

win/bin/

win/build/

src/main.cpp

```
//=====
#include <iostream>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    string m_name;
    int m_code;

    cout << "\n### Saisir vos identifiants:\n";
    cout << "Nom: "; cin >> m_name;
    cout << "Code: "; cin >> m_code;

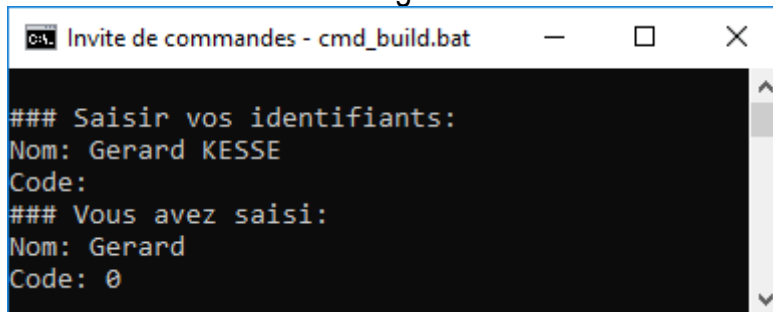
    cout << "\n### Vous avez saisi:\n";
    cout << "Nom: " << m_name << "\n";
    cout << "Code: " << m_code << "\n";
    return 0;
}
//=====
```

Problème :

Problème de gestion d'espace lors de la saisie

Nom : Gerard KESSE

Code : la saisie du code est ignorée



Gérer le problème d'espace lors de la saisie de données

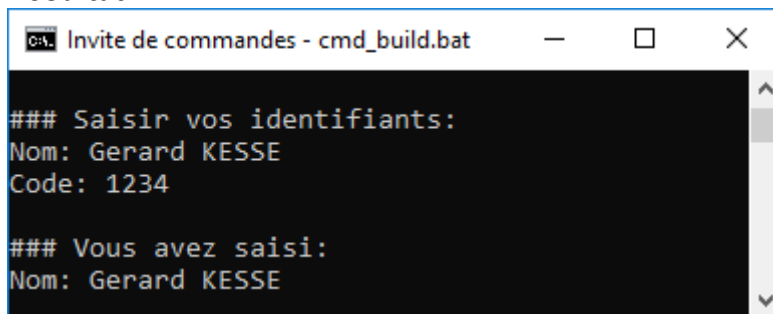
Objectif :

Gérer le problème d'espace lors de la saisie de données à partir du clavier.

Implémentation :

Récupérer les identifiants d'une personne saisis à partir du clavier.

Récupérer le nom (m_name) et le code d'entrée (m_code) de la personne.

Résultat :

```
### Saisir vos identifiants:
Nom: Gerard KESSE
Code: 1234

### Vous avez saisi:
Nom: Gerard KESSE
```

Dossier projet :

src/main.cpp

win/bin/

win/build/

src/main.cpp

```
//=====
#include <iostream>
#include <string>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    string m_cin;
    string m_name;
    int m_code;

    cout << "\n### Saisir vos identifiants:\n";
    cout << "Nom: "; getline(cin, m_cin); m_name =
m_cin;
    cout << "Code: "; getline(cin, m_cin); m_code =
stoi(m_cin);

    cout << "\n### Vous avez saisi:\n";
    cout << "Nom: " << m_name << "\n";
    cout << "Code: " << m_code << "\n";
    return 0;
}
//=====
```


Gérer les conditions

Condition If (Si)

Objectif :

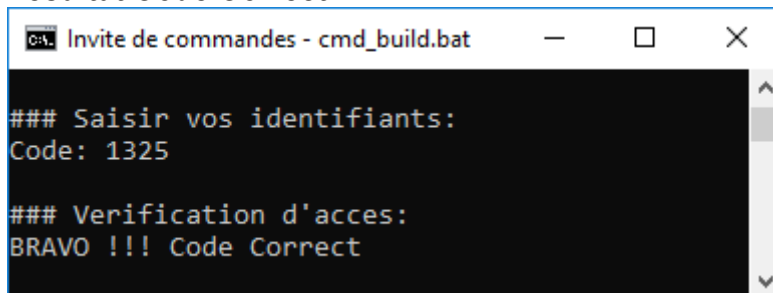
Exécuter une instruction Si une condition est vérifiée.

Implémentation :

Récupérer le code d'accès (m_code) d'une personne.

Afficher un message (BRAVO !!! Code Correct) si le code d'accès est correct.

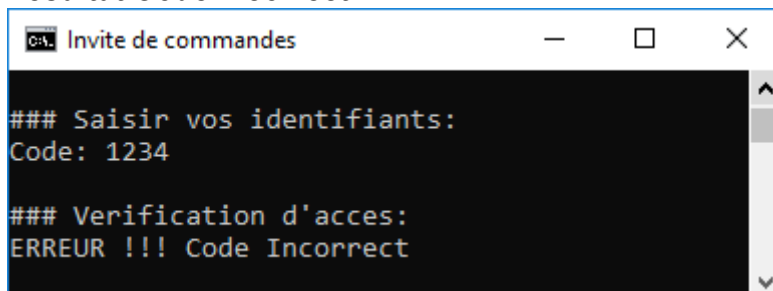
Résultat Code Correct :



```
### Saisir vos identifiants:
Code: 1325

### Verification d'accès:
BRAVO !!! Code Correct
```

Résultat Code Incorrect :



```
### Saisir vos identifiants:
Code: 1234

### Verification d'accès:
ERREUR !!! Code Incorrect
```

Dossier projet :

src/main.cpp

win/bin/

win/build/

src/main.cpp

```
//=====
#include <iostream>
#include <string>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    string m_cin;
    int m_code;
    int ACCESS_CODE = 1325;

    cout << "\n### Saisir vos identifiants:\n";
    cout << "Code: "; getline(cin, m_cin); m_code =
stoi(m_cin);

    cout << "\n### Verification d'accès:\n";
    if(m_code == ACCESS_CODE) {
        cout << "BRAVO !!! Code Correct\n";
        return 0;
    }
    cout << "ERROR !!! Code Incorrect\n";
    return 0;
}
//=====
```

Condition Else (Sinon)

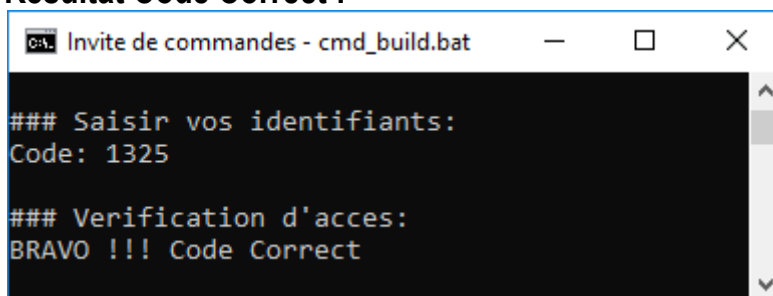
Objectif :

Exécuter une instruction Si une condition est vérifiée,
Sinon exécuter l'autre instruction.

Implémentation :

Récupérer le code d'accès (m_code) d'une personne.
Afficher un message (BRAVO !!! Code Correct) si le code d'accès est correct.
Sinon afficher le message (ERREUR !!! Code Incorrect).

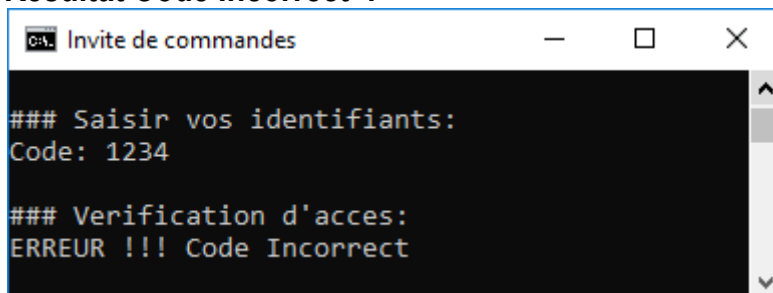
Résultat Code Correct :



```
### Saisir vos identifiants:
Code: 1325

### Verification d'accès:
BRAVO !!! Code Correct
```

Résultat Code Incorrect :



```
### Saisir vos identifiants:
Code: 1234

### Verification d'accès:
ERREUR !!! Code Incorrect
```

Dossier projet :

src/main.cpp
win/bin/
win/build/

src/main.cpp

```
//=====
#include <iostream>
#include <string>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    string m_cin;
    int m_code;
    int ACCESS_CODE = 1325;

    cout << "\n### Saisir vos identifiants:\n";
    cout << "Code: "; getline(cin, m_cin); m_code =
stoi(m_cin);

    cout << "\n### Verification d'accès:\n";
    if(m_code == ACCESS_CODE) {
        cout << "BRAVO !!! Code Correct\n";
        return 0;
    }
    cout << "ERROR !!! Code Incorrect\n";
    return 0;
}
//=====
```

Condition Else If (Sinon Si)

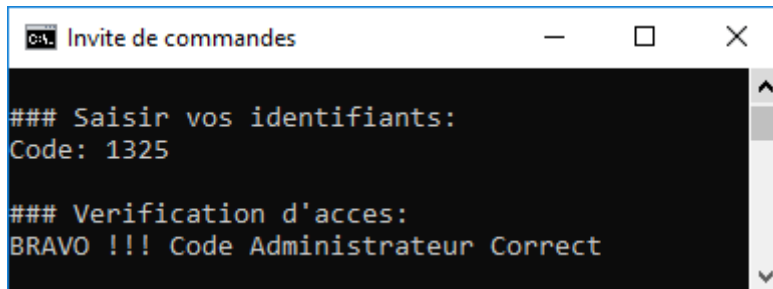
Objectif :

Exécuter une instruction Si une condition est vérifiée,
Sinon Si la condition suivante est vérifiée, exécuter l'autre instruction.

Implémentation :

Récupérer le code d'accès (m_code) d'une personne.
Si le code d'accès correspond au code d'accès administrateur,
Afficher un message (BRAVO !!! Code Administrateur Correct).
Sinon Si le code d'accès correspond au code d'accès utilisateur,
Afficher le message (BRAVO !!! Code Utilisateur Correct).
Dans le Cas Contraire (Sinon),
Afficher le message (ERREUR !!! Code Incorrect).

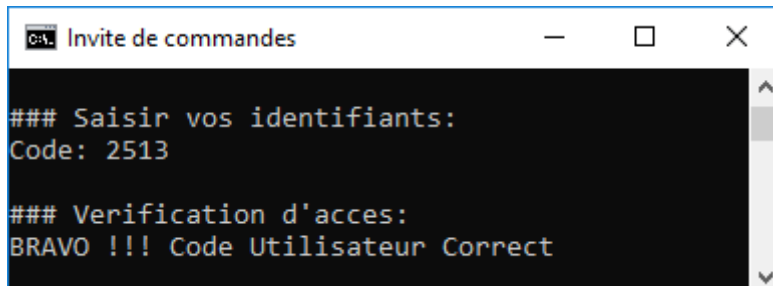
Résultat Code Administrateur Correct :



```
### Saisir vos identifiants:
Code: 1325

### Verification d'accès:
BRAVO !!! Code Administrateur Correct
```

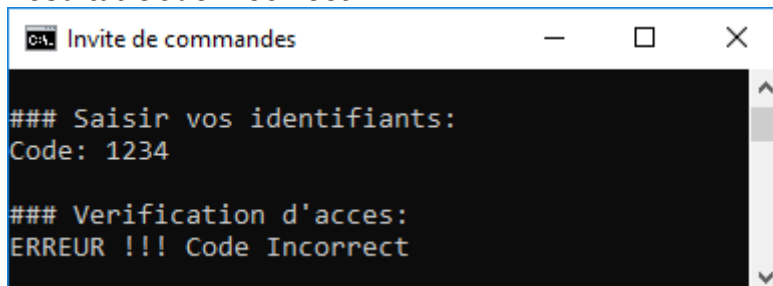
Résultat Code Utilisateur Correct :



```
### Saisir vos identifiants:
Code: 2513

### Verification d'accès:
BRAVO !!! Code Utilisateur Correct
```

Résultat Code Incorrect :



```
### Saisir vos identifiants:
Code: 1234

### Verification d'accès:
ERREUR !!! Code Incorrect
```

Dossier projet :

src/main.cpp
win/bin/
win/build/

src/main.cpp

```
//=====
#include <iostream>
#include <string>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    string m_cin;
    int m_code;
    int ADMIN_CODE = 1325;
    int USER_CODE = 2513;

    cout << "\n### Saisir vos identifiants:\n";
    cout << "Code: "; getline(cin, m_cin); m_code =
stoi(m_cin);

    cout << "\n### Verification d'accès:\n";
    if(m_code == ADMIN_CODE) {
        cout << "BRAVO !!! Code Administrateur
Correct\n";
    }
    else if(m_code == USER_CODE) {
        cout << "BRAVO !!! Code Utilisateur
Correct\n";
    }
    else {
        cout << "ERREUR !!! Code Incorrect\n";
    }
    return 0;
}
//=====
```

Gérer les boucles

Boucle For (Pour Tout)

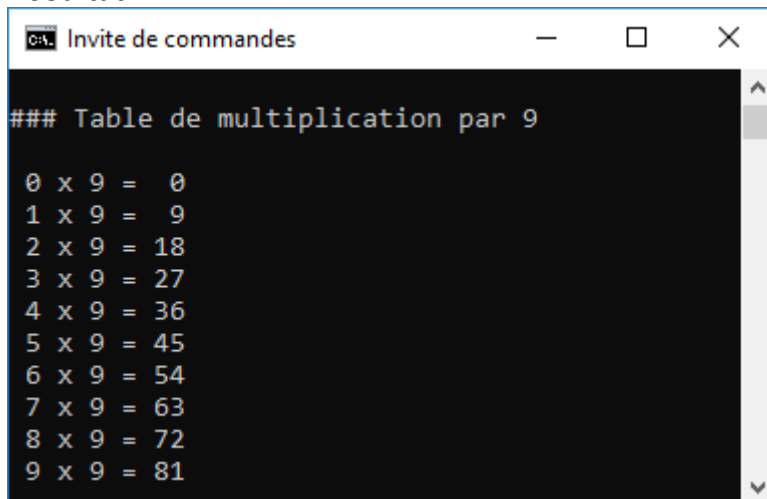
Objectif :

Exécuter une instruction en boucle.

Implémentation :

Créer une table de multiplication par 9.

Les lignes allant de (0 à 10).

Résultat :

```
### Table de multiplication par 9
0 x 9 = 0
1 x 9 = 9
2 x 9 = 18
3 x 9 = 27
4 x 9 = 36
5 x 9 = 45
6 x 9 = 54
7 x 9 = 63
8 x 9 = 72
9 x 9 = 81
```

Dossier projet :

src/main.cpp

win/bin/

win/build/

src/main.cpp

```
//=====
#include <iostream>
#include <iomanip>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    int ROW_MAX = 10;
    int FACTOR = 9;

    cout << "### Table de multiplication par 9\n\n";

    for(int i = 0; i <= ROW_MAX; i++) {
        int m_data = i * FACTOR;
        cout << setw(2) << i << " x ";
        cout << FACTOR << " = ";
        cout << setw(2) << m_data << "\n";
    }
    return 0;
}
//=====
```


Boucle While (Tant Que)

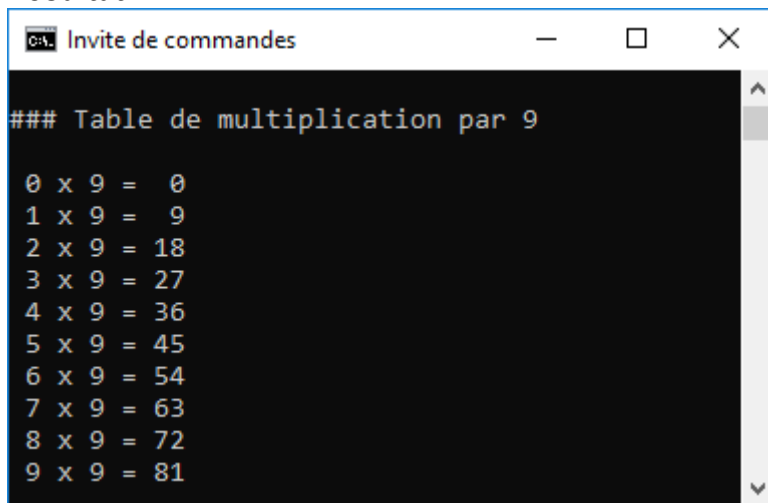
Objectif :

Exécuter une instruction en boucle.

Implémentation :

Créer une table de multiplication par 9.

Les lignes allant de (0 à 10).

Résultat :

```
### Table de multiplication par 9
0 x 9 = 0
1 x 9 = 9
2 x 9 = 18
3 x 9 = 27
4 x 9 = 36
5 x 9 = 45
6 x 9 = 54
7 x 9 = 63
8 x 9 = 72
9 x 9 = 81
```

Dossier projet :

src/main.cpp

win/bin/

win/build/

src/main.cpp

```
//=====
#include <iostream>
#include <iomanip>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    int ROW_MAX = 10;
    int FACTOR = 9;

    cout << "### Table de multiplication par 9\n\n";

    int i = 0;

    while(i <= ROW_MAX) {
        int m_data = i * FACTOR;
        cout << setw(2) << i << " x ";
        cout << FACTOR << " = ";
        cout << setw(2) << m_data << "\n";
        i++;
    }
    return 0;
}
//=====
```

Boucle Do While (Faire Tant Que)

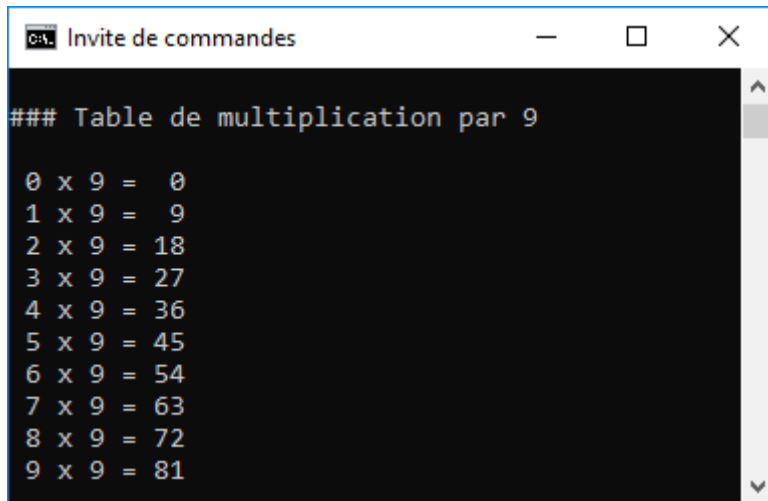
Objectif :

Exécuter une instruction en boucle.

Implémentation :

Créer une table de multiplication par 9.

Les lignes allant de (0 à 10).

Résultat :

```
### Table de multiplication par 9

0 x 9 = 0
1 x 9 = 9
2 x 9 = 18
3 x 9 = 27
4 x 9 = 36
5 x 9 = 45
6 x 9 = 54
7 x 9 = 63
8 x 9 = 72
9 x 9 = 81
```

Dossier projet :

src/main.cpp

win/bin/

win/build/

src/main.cpp

```
//=====
#include <iostream>
#include <iomanip>
//=====
using namespace std;
//=====
int main(int argc, char** argv) {
    int ROW_MAX = 10;
    int FACTOR = 9;

    cout << "### Table de multiplication par 9\n\n";

    int i = 0;

    do {
        int m_data = i * FACTOR;
        cout << setw(2) << i << " x ";
        cout << FACTOR << " = ";
        cout << setw(2) << m_data << "\n";
        i++;
    } while(i <= ROW_MAX);
    return 0;
}
//=====
```