

Algorithmique & Langage C

- Modalités
- Programme du Cours
- Environnement pédagogique



Modalités du cours

- 2 séances x 2 heures par semaine de Cours / TD
- 30 heures de TP
- Evaluation
 - DS + Quizz
 - 1 partiel + 2^e session



Programme du cours

- Information sur le fonctionnement de Visual Studio
- Convention de codage - indentation
- Instructions E/S console (printf, scanf)

- Ecriture d'un premier programme simple pour faire une démo interactive
- Débogage point arrêt - variables - mémoire



Programme du cours

CODAGE

- Introduction au binaire et au codage des entiers.
- Bits, octets (bytes)
- Conversion (décimal -> binaire, binaire -> décimal)
- Changements de bases courants (ex binaire vers hexa)
- Codage de caractères
- Représentation des entiers, signés et non signés
- Arithmétique binaire (addition – soustraction)
- Opérations de bit à bit



Programme du cours

- **Type simples de données :**
 - Numériques (entiers, décimaux), dynamiques des types
 - Alphanumériques caractères, chaînes de caractères
 - Booléennes (vrai, faux)
- **Variables** (concept, déclaration, assignation)
- **Opérateurs** (unaire, binaire, priorités)
- **Expressions** (évaluation, type)



Programme du cours

Structures conditionnelles : (exercices en pseudocode et en C)

- If then else
- Imbrications des structures conditionnelles
- Switch case

Structures Itératives (exercices en pseudocode et langage C)

- Pour
- Tant que
- Faire .. tant que



Programme du cours

- Introduction aux tableaux
- Introduction aux (procédures et) fonctions
- Tests unitaires
- Fonctions récursives
- Fonctions récursives terminales

Programme du cours

Etudes de méthodes de tri

- tri à bulles
- tri par sélection ordinaire
- tri par insertion séquentielle
- tri par insertion dichotomique
- tri par tas
- tri par fusion
- tri rapide
- ...



Programme du cours

- Structures
- Pointeurs
- Allocations dynamiques



Programme du cours

Fonctions :

- Passage de paramètres dans les fonctions par valeurs ou par références.
- Retour de fonctions
- Portée des variables
- Tableaux à plusieurs dimensions et appels de fonction
- Jeu de la vie



Programme du cours

Structures de données avancées :

- Listes Chaînées Unilatères et Bilatères
- Piles version tableau et liste chaînée
- Files version tableau et liste chaînée
- Arbres : concepts
- Arbres Binaires de Recherche (ABR)

Ouvrages de référence :

Pour les notions
d'algorithmique :



Ouvrages de référence :

Pour les notions liées au
codage en C :



Disponible sur la librairie en ligne !



Environnement Pédagogique

- Ordinateur portable sous **Windows**, WSL2, Linux ou MacOS
- Equipé d'un IDE
 - **Microsoft Visual Studio Community 2019** ou 2022 Preview (Release en novembre 2021)
 - Visual Studio Code
- Build
 - **Solution Visual Studio *.sln** + MSBUILD + MSVC sous Windows
 - CMAKE + gcc ou Clang/LLVM
- Plateformes en ligne
 - **Teams** pour la mise à disposition de certains supports pédagogiques
 - **GitHub** pour la publication et le partage sur des dépôts de code



Environnement Pédagogique

- [prise en charge de Clang/LLVM dans les projets Visual Studio | Microsoft Docs](#)
- [prise en charge de Clang/LLVM dans Visual Studio projets CMake | Microsoft Docs](#)

Consulter le document

Mise en place envt VSC-CMAKE-CLANG

Pour obtenir les liens de référence vers les différents outils

Pascal RICQ 2021 - réf PRG1000-intro-v1.0



Microsoft Teams

The screenshot shows the Microsoft Teams interface for the 'CS13 AlgoC 65' team. The left sidebar contains navigation icons for Activity, Conversation, Équipes (selected), Devoirs, Calendrier, Appels, Fichiers, and Applications. The main content area is divided into three sections: 'Actualités' (News), 'Activité' (Activity), and 'Liens rapides' (Quick links). The 'Actualités' section features a post about 'Cours Algorithmique et Langage C en CS13' by Pascal RICO, dated 27 juillet. The 'Activité' section shows three recent documents: 'SitePages Cours-Algorithmique-et-Langage-C-en-CS13' (modified 27 juillet), 'SitePages 5hk3c5ck' (added 27 juillet), and 'Le nouveau groupe CS13 AlgoC 65 est prêt' (sent 27 juillet). The 'Liens rapides' section includes links to 'En savoir plus sur un site d'équipe' and 'Découvrez comment ajouter une page'. The 'Documents' section on the right lists various folders and files, including 'General', 'TD1', 'TD2', 'TD3', 'TD4', 'TP1', and 'TP2', all dated 27 juillet.

Rechercher

Toutes les équipes

CS13 AlgoC 65

Général

Intervenants

TD1

TD2

TD3

TD4

TP1

TP2

TP3

TP4

TP5

Actualités

PRO Git

Cours Algorithmique et Langage C en CS13

L'espace Junia AlgoC CS13 rassemble vos dépôts de code créés pour ce cours...

Pascal RICO 27 juillet

Activité

SitePages

Cours-Algorithmique-et-Langage-C-en-CS13

Pascal RICO Modifié 27 juillet

SitePages

5hk3c5ck

Pascal RICO Ajouté 27 juillet

Le nouveau groupe CS13 AlgoC 65 est prêt

Bienvenue dans le groupe CS13 AlgoC 65. Utilisez le groupe pour partager des idées, des fichiers et

CS13 AlgoC 65 +1 Envoyé 27 juillet

Liens rapides

En savoir plus sur un site d'équipe

Découvrez comment ajouter une page

Documents

Tous les documents

Nom	Modifié
General	27 juillet
TD1	27 juillet
TD2	27 juillet
TD3	27 juillet
TD4	27 juillet
TP1	27 juillet
TP2	27 juillet



GitHub

<https://github.com/JuniaCIR1>

A screenshot of the GitHub profile page for the organization "Junia Algorithmic & C Course". The page shows the organization's name, description, location (Lille - France), website, and email. Below this, there are tabs for Overview, Repositories (3), Packages, People (3), Teams (5), and Projects. The Repositories tab is selected, showing a list of three repositories: "git--pascalricq" (Private), "BinaryCalc", and "SortMethods" (Private). Each repository entry includes its name, description, language (C), star count, fork count, and update date. On the right side, there are sections for "People" (showing three avatars) and "Top languages" (showing C).



GitHub

<https://github.com/JuniaCSI3>

The screenshot shows the GitHub profile page for the user JuniaCSI3. The page header includes navigation links: Pull requests, Issues, Marketplace, and Explore. The user's profile section displays the Junia logo, the name "Junia AlgoC CSI3", and the description "Algorithmic and C programming Language Course". Below this, there are tabs for Overview, Repositories (1), Packages, People (1), Teams, Projects, and Settings. The main content area shows a "Pinned" section with the message "JuniaCSI3 doesn't have any pinned public repositories yet." and a "Customize your pins" link. Below this is a "Repositories" section with a search bar, filters for "Type" and "Sort", and a "New" button. A single repository, "github-starter-course", is listed as a "Private template" created by GitHub Classroom, with 0 stars, 0 forks, 0 issues, and 0 pull requests, and was updated on 10 Jun. A "View all repositories" link is at the bottom.





Pré-requis
pour le cours :

VS Community

Option

VS Code pour
les autres
travaux



Visual Studio 2019

Version 16.6

[Notes de publication >](#)

IDE (environnement de développement intégré) complet pour Android, iOS, Windows, le web et le cloud

[Comparaison des éditions >](#)
[Installation hors connexion >](#)

Communauté

Puissant IDE, gratuit pour les étudiants, les contributeurs open source et les développeurs indépendants

Téléchargement gratuit ↓

Professional

IDE Professional adapté aux petites équipes

Essai gratuit ↓

Entreprise




Solution de bout en bout et scalable pour les équipes de toutes tailles

Essai gratuit ↓

Visual Studio Preview

[Notes de publication >](#)

Bénéficiez d'un accès anticipé aux fonctionnalités les plus récentes qui ne sont pas encore disponibles dans la version principale
[En savoir plus >](#)





Visual Studio Code

[Notes de publication >](#)

L'éditeur de code rapide, gratuit et open source qui s'adapte à vos besoins

Téléchargement gratuit ↓

En téléchargeant et en utilisant Visual Studio Code, vous acceptez les [termes du contrat de licence](#) et la [déclaration de confidentialité](#).



Visual Studio pour Mac

Version 8.6

[Notes de publication >](#)

Développez des applications et des jeux pour iOS, Android et le web en utilisant .NET

Téléchargement gratuit ↓

[Procédure d'activation de votre licence.](#)Sélectionnez votre édition après l'installation. L'installation de Visual Studio pour Mac nécessite une connexion Internet active.

Visual Studio

Accès aux logiciels Microsoft sur le site Azure Portal Education :

<https://aka.ms/type3>

- Il faut disposer d'un compte académique sur Microsoft 365
- Attendre que les logiciels soient « provisionnés » pour votre compte
- En cas de problème, contacter le support 2345



Visual Studio

Sur Azure Portal Education :

- Récupérer la clé d'activation de Visual Studio Enterprise sur ce site avant la fin de la période d'essai et activer le logiciel



Microsoft Azure

Accueil > Education - Logiciels

Education - Logiciels

APERÇU

Rechercher (Ctrl+ /)

Vue d'ensemble

Templates

Formation

Logiciels

BESOIN D'AIDE ?

FAQ Étudiant

Rechercher

Catégorie du produit : All

Type de système : 64 bits, Service

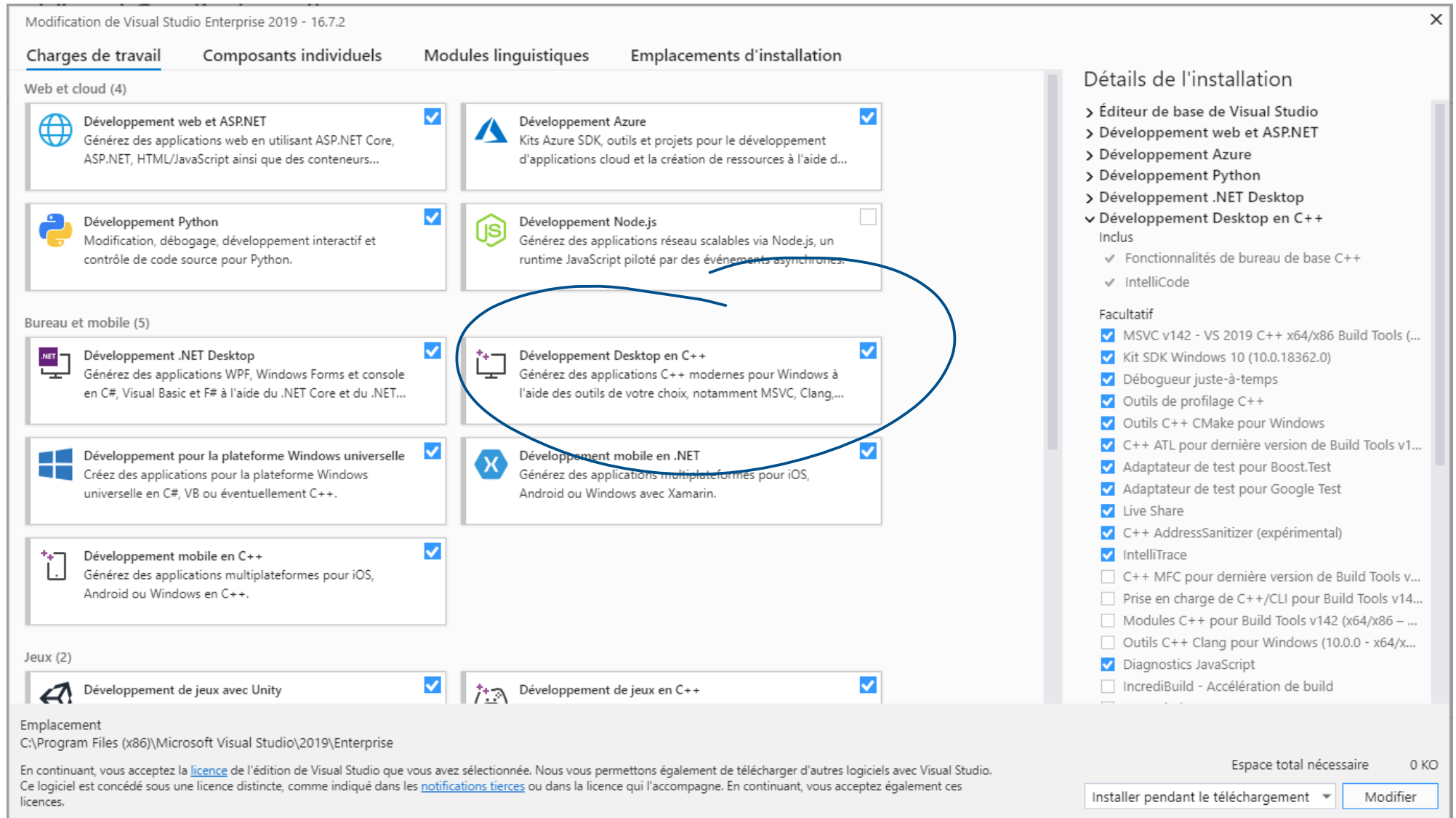
Langue du produit : Français, Multilingue

94 Éléments

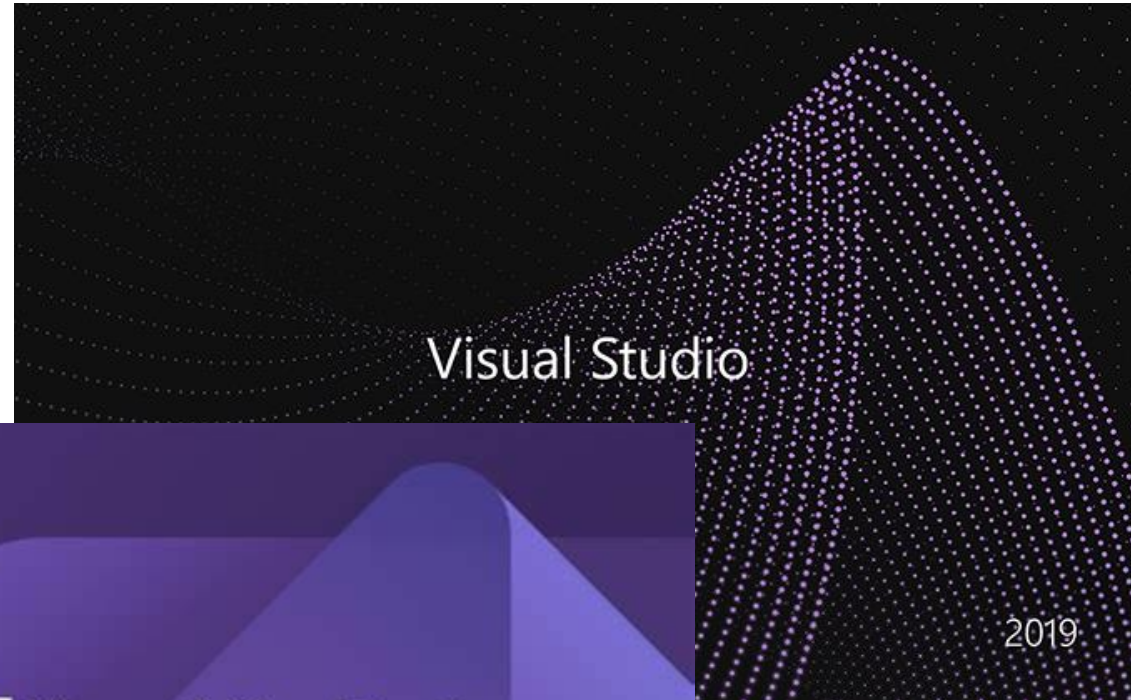
NOM	CATÉGORIE DU PRODUIT	TYPE DE SYSTÈME	LANGUE
Windows Server 2019 (updated Marc...	Operating System	64 bits	Français
Visual Studio Enterprise 2019	Developer Tools	64 bits	Multilingue
Windows Server 2019 64 bits	Système d'exploitation	64 bits	Français
WintellectNow	Services de formation	Service	Multilingue
Microsoft Store	Outils de productivité	Service	Multilingue
Visual Studio Code	Outils de développement	64 bits	Multilingue
SQL Server 2017 Enterprise	Base de données	64 bits	Français
Project Professionnel 2019	Outils de productivité	64 bits	Français
Access 2016	Base de données	64 bits	Français
Advanced Threat Analytics (ATA) vers...	Sécurité	64 bits	Multilingue
Advanced Threat Analytics (ATA) Vers...	Sécurité	64 bits	Multilingue
Agents for Visual Studio 2019 (versio...	Developer Tools	64 bits	Multilingue
Agents for Visual Studio 2019 (versio...	Developer Tools	64 bits	Multilingue
Azure DevOps Server 2019.0.1 (x64) -...	Productivity Tools	64 bits	Multilingue
Azure DevOps Server 2019.0.1 (x64) -...	Productivity Tools	64 bits	Multilingue
Azure DevOps Server Express 2019.0....	Productivity Tools	64 bits	Multilingue
Azure DevOps Server Express 2019.0....	Productivity Tools	64 bits	Multilingue
Agents pour Visual Studio 2017 Test...	Outils de développement	64 bits	Multilingue

Visual Studio Installer :

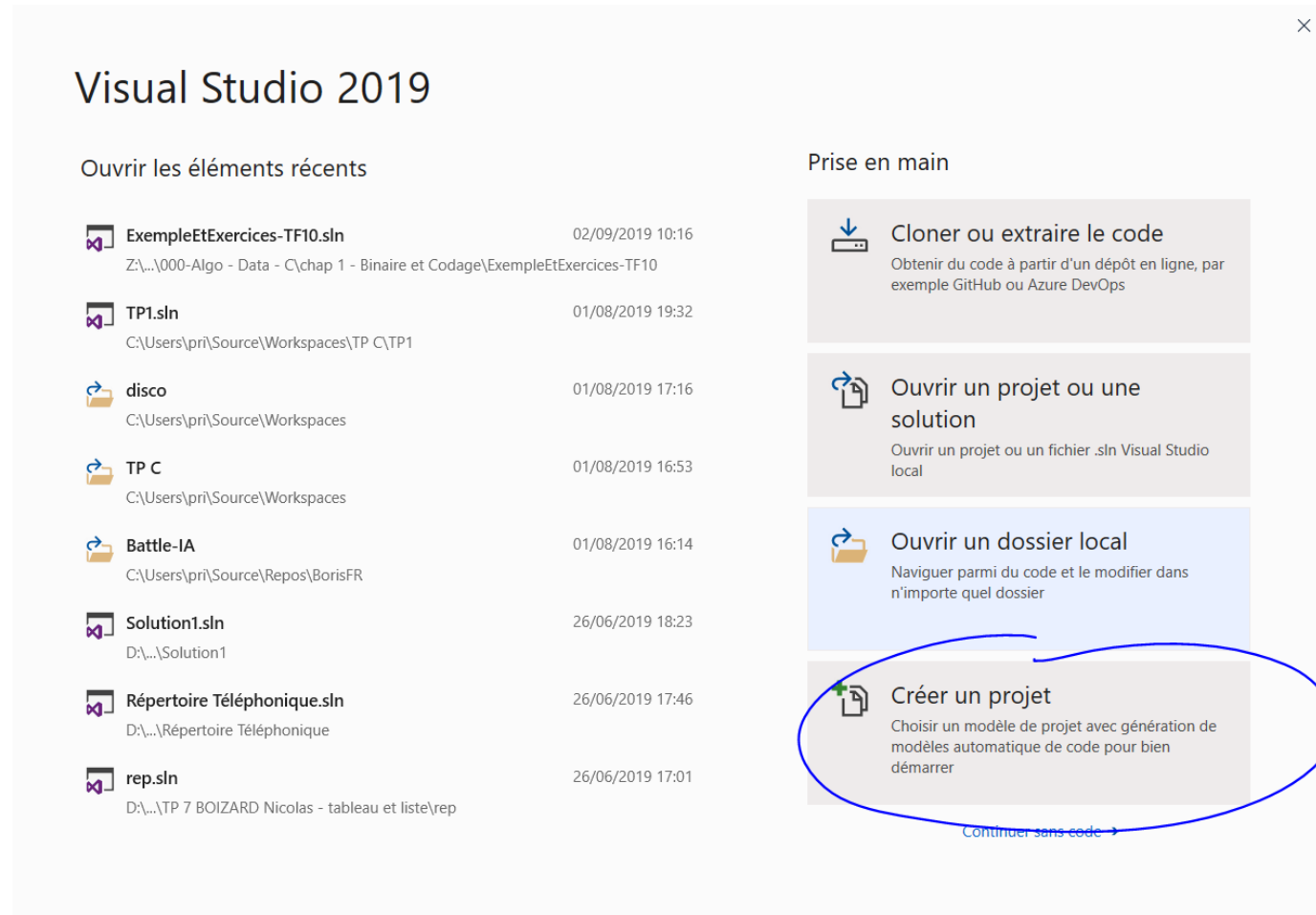
Choisir l'option
Développement
Desktop en C++



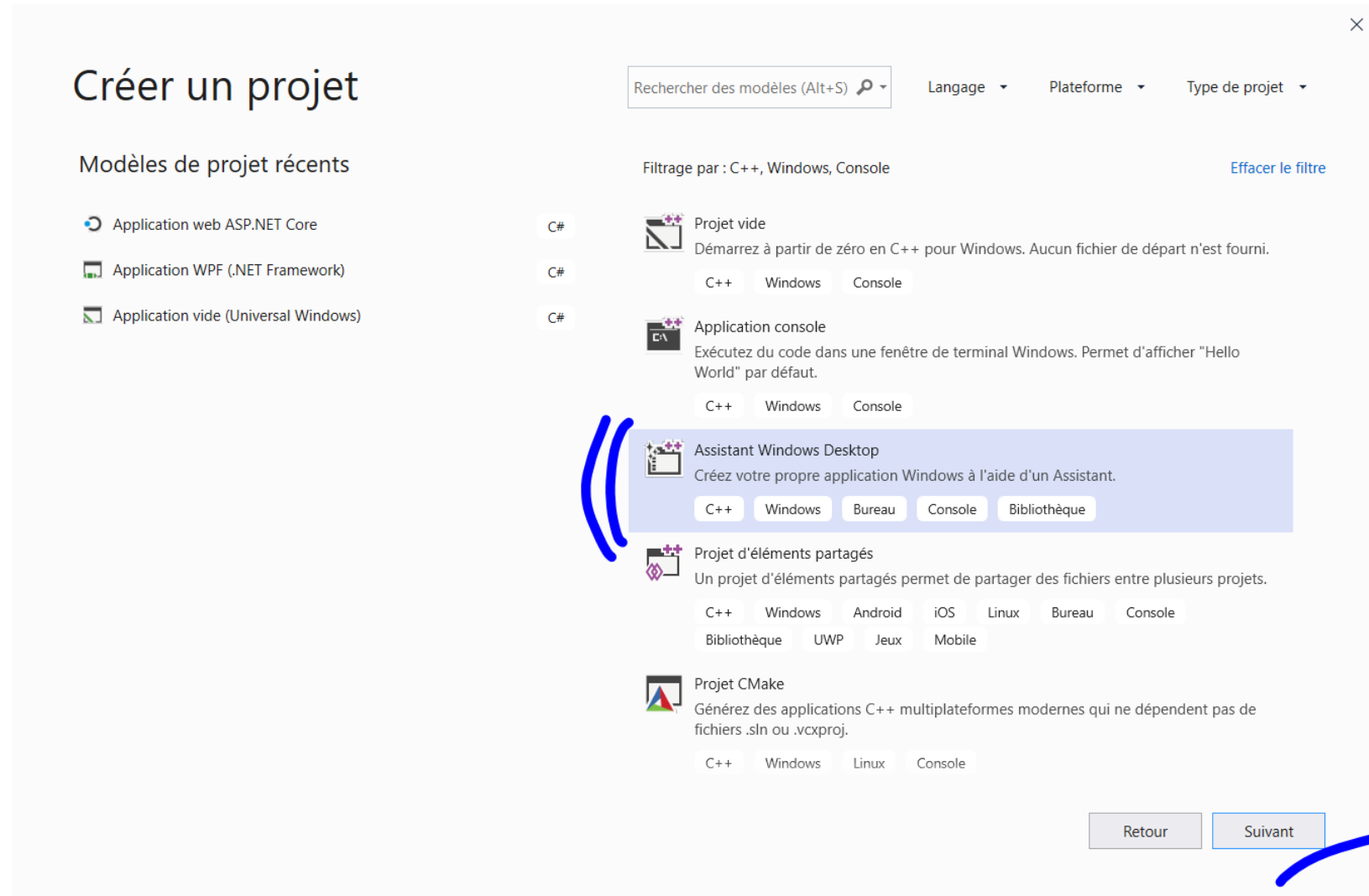
Microsoft Visual Studio



Visual Studio de Microsoft



Visual Studio de Microsoft





Visual Studio de Microsoft

×

Configurer votre nouveau projet

Assistant Windows Desktop C++ Windows Bureau Console Bibliothèque

Nom du projet

Emplacement

...

Nom de la solution ⓘ

☐ Placer la solution et le projet dans le même répertoire

Retour

Créer

Visual Studio de Microsoft


Projet Windows Desktop

Type d'application

Application console (.exe)

Options supplémentaires :

- ☒ Projet vide
- ☐ En-tête précompilé
- ☐ Exporter les symboles
- ☐ En-têtes MFC

 Conseil : Vous pouvez également utiliser le modèle de projet vide pour créer ce type de projet.

OK Annuler

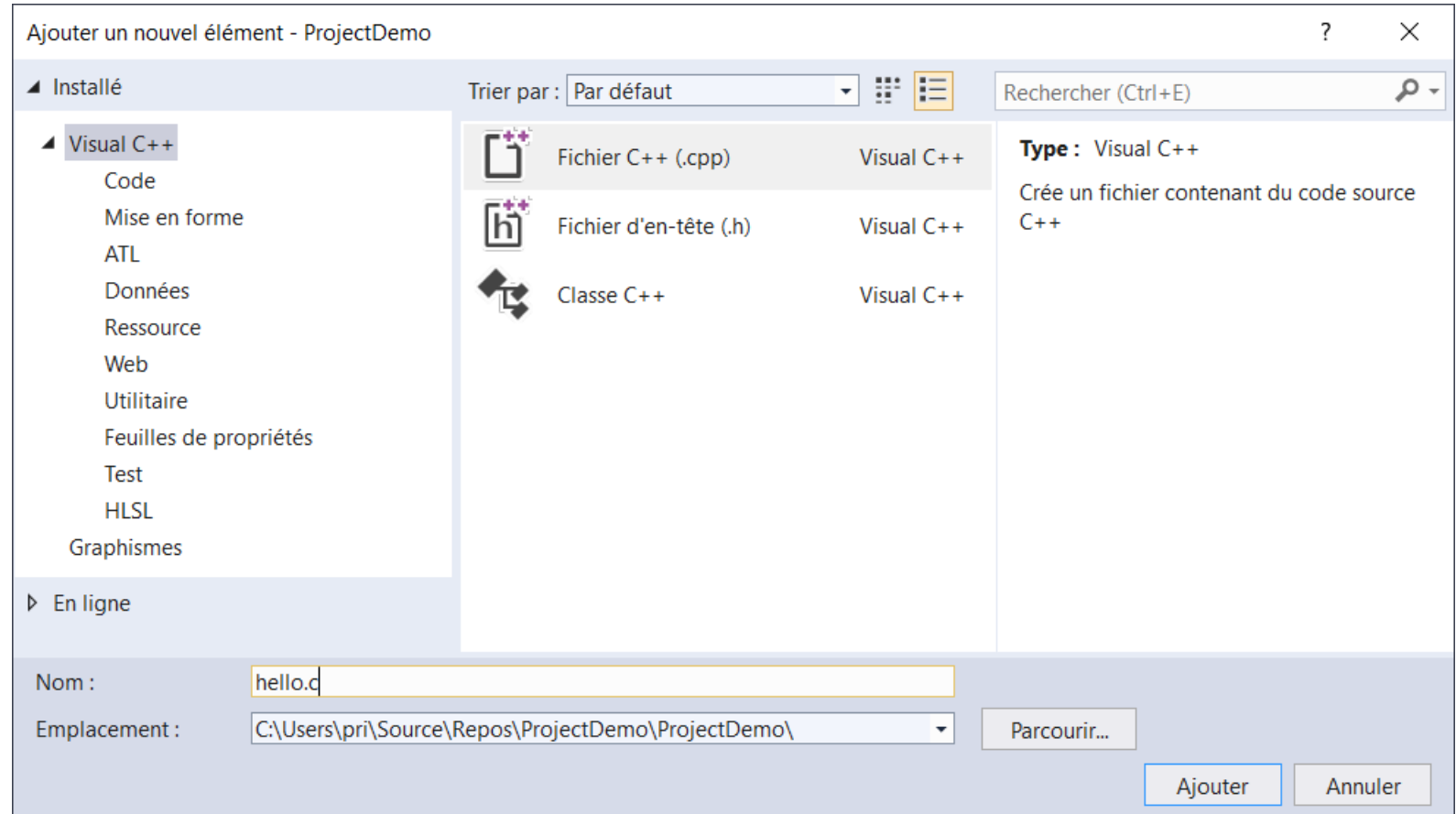
Visual Studio de Microsoft

Dans la fenêtre « Explorateur de solutions » ajoutez votre premier fichier source C en faisant un clic-droit sur le dossier Source Files et en sélectionnant:

- Ajouter un nouvel élément puis
- C++ File (.cpp)

Attention, dans le champ Nom du fichier, vous devez entrer un nom de fichier avec l'extension « .c » et non « .cpp »

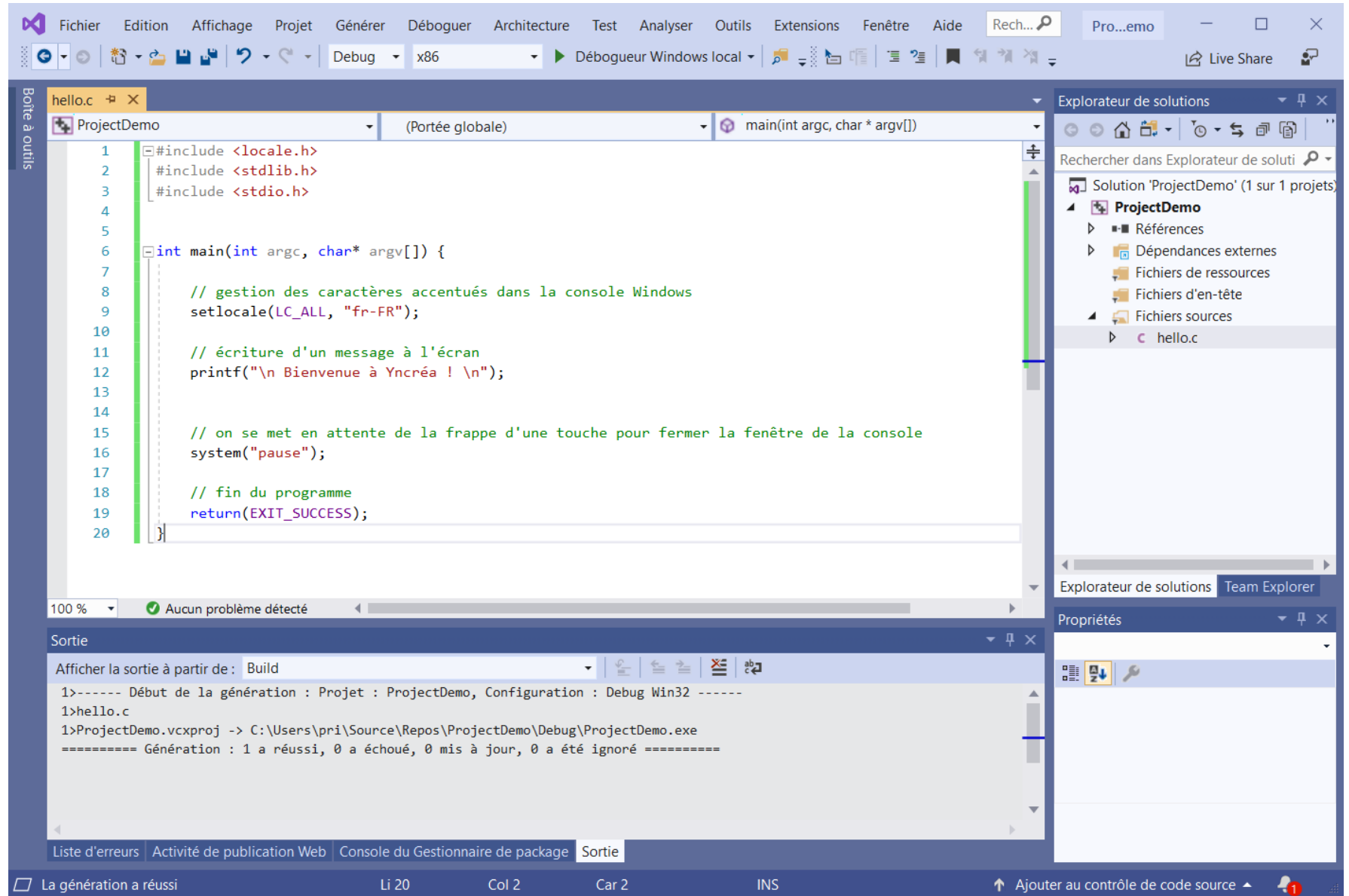
Par exemple, nommez le fichier source « hello.c » et cliquez sur Add.



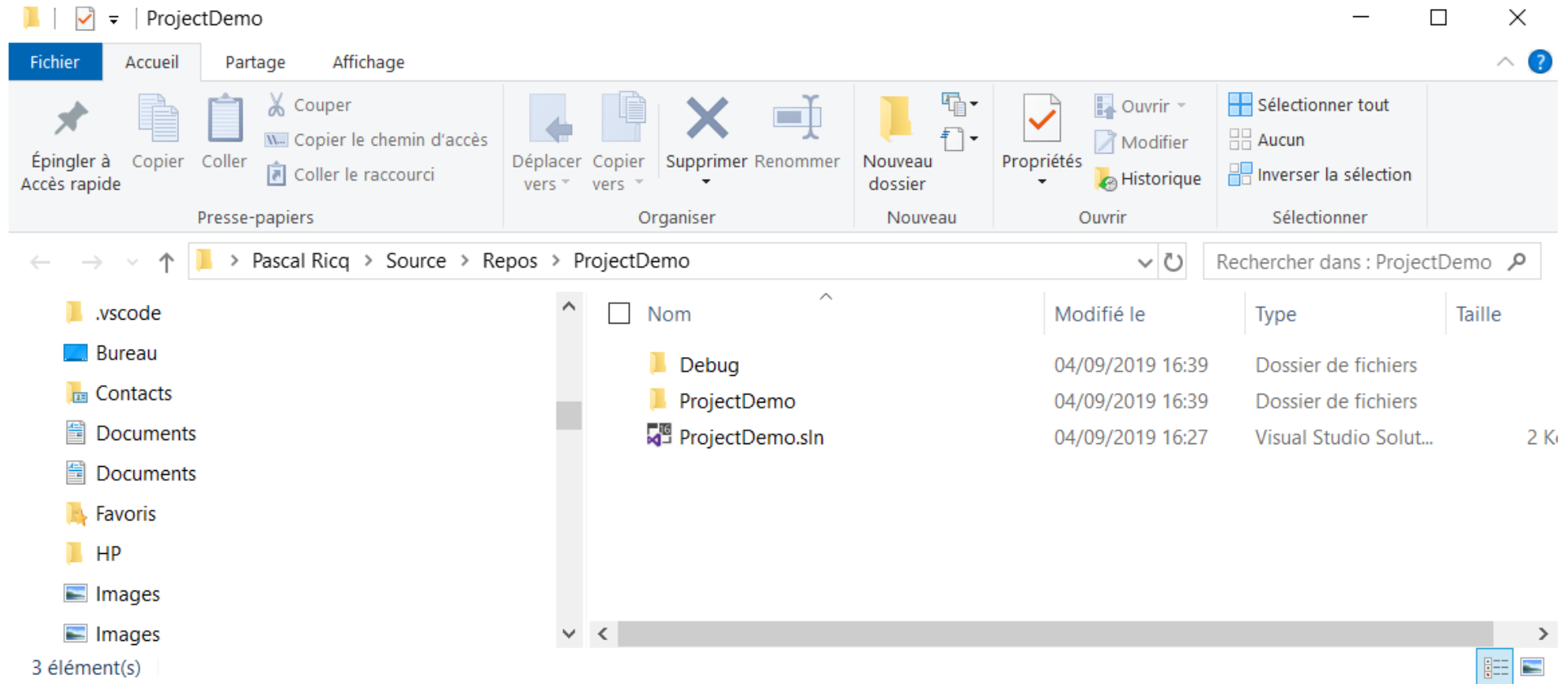
Dans la fenêtre principale, on peut saisir le code C du fichier hello.c

La compilation et l'exécution du programme peuvent se faire en cliquant sur l'icône verte Débogueur Windows Local ou dans le menu Générer/Démarrer le débogage.

Si le programme a été juste modifié, il faut accepter de faire la recompilation (générer).



Exploration du dossier de la "solution" qui contient un projet "ProjectDemo"

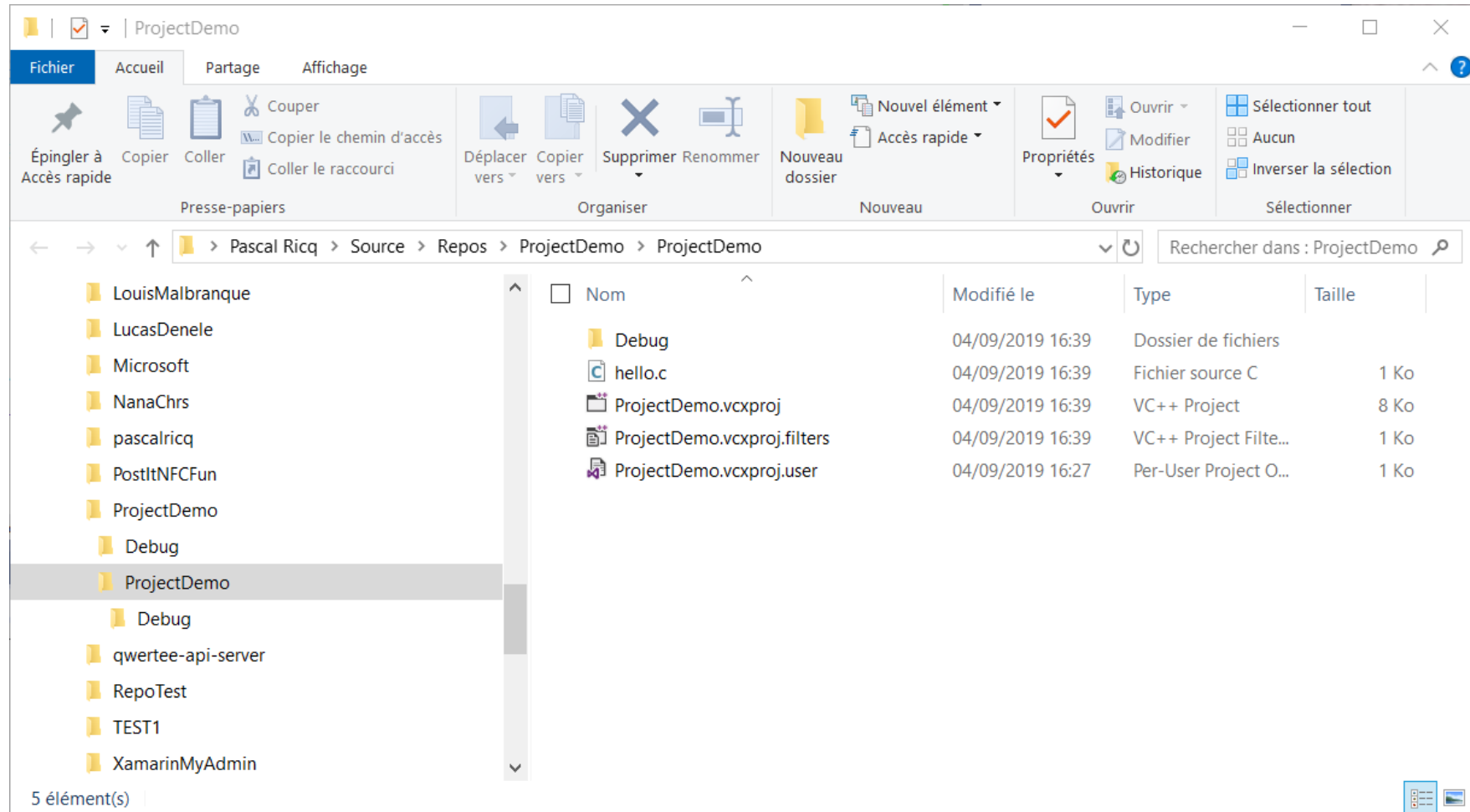


The screenshot shows a Windows File Explorer window titled "ProjectDemo". The address bar indicates the path: "Pascal Ricq > Source > Repos > ProjectDemo". The left sidebar shows the "This PC" view with various folders like .vscode, Bureau, Contacts, Documents, Favoris, HP, Images, and Images. The main pane displays a list of files and folders:

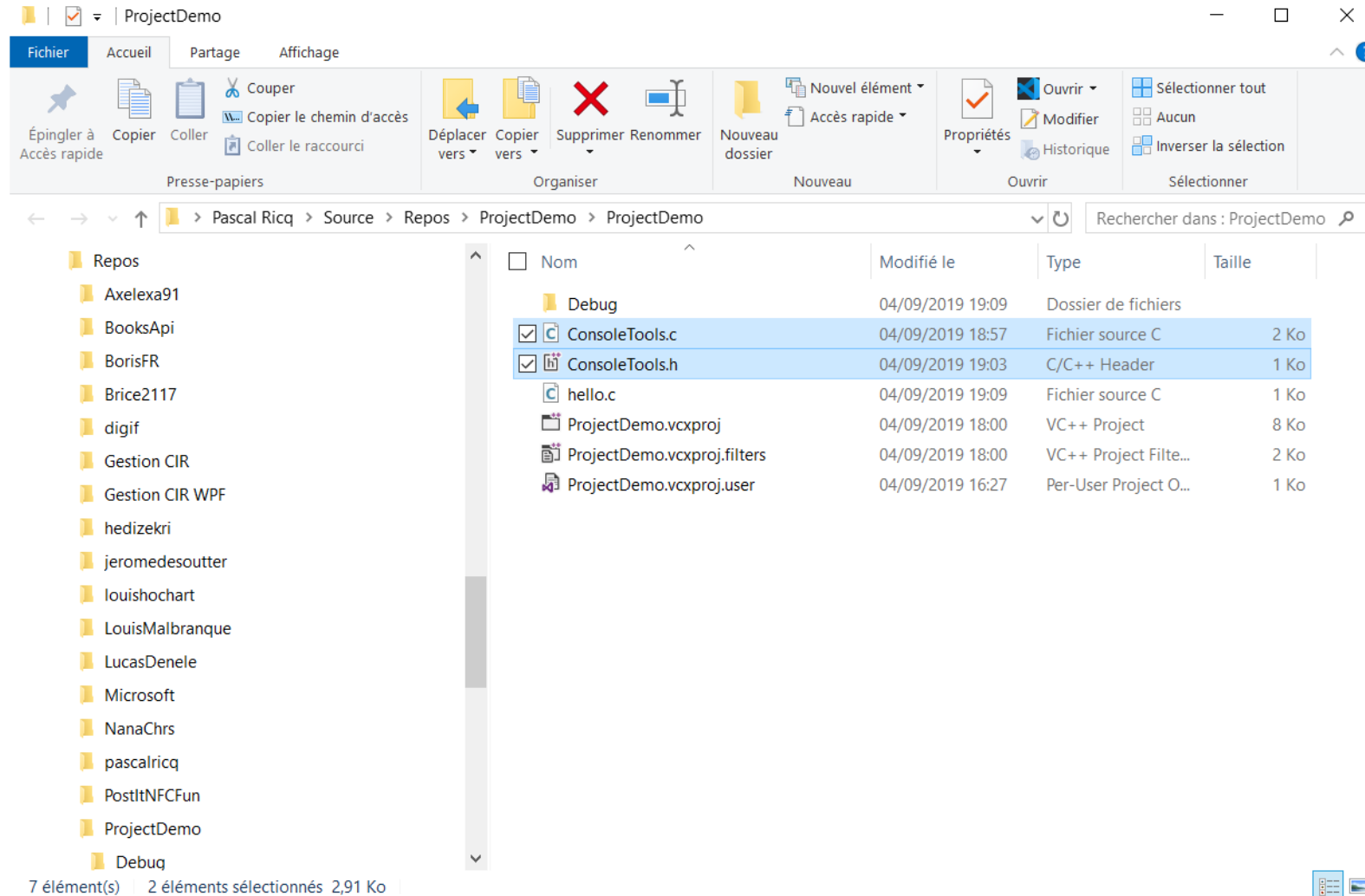
Nom	Modifié le	Type	Taille
Debug	04/09/2019 16:39	Dossier de fichiers	
ProjectDemo	04/09/2019 16:39	Dossier de fichiers	
ProjectDemo.sln	04/09/2019 16:27	Visual Studio Solut...	2 Ki

The bottom status bar indicates "3 élément(s)".

Exploration du dossier du "projet" qui contient les fichiers sources du projet



On peut y ajouter des d'autres fichiers et les ajouter au projets



ProjectDemo

Fichier Accueil Partage Affichage

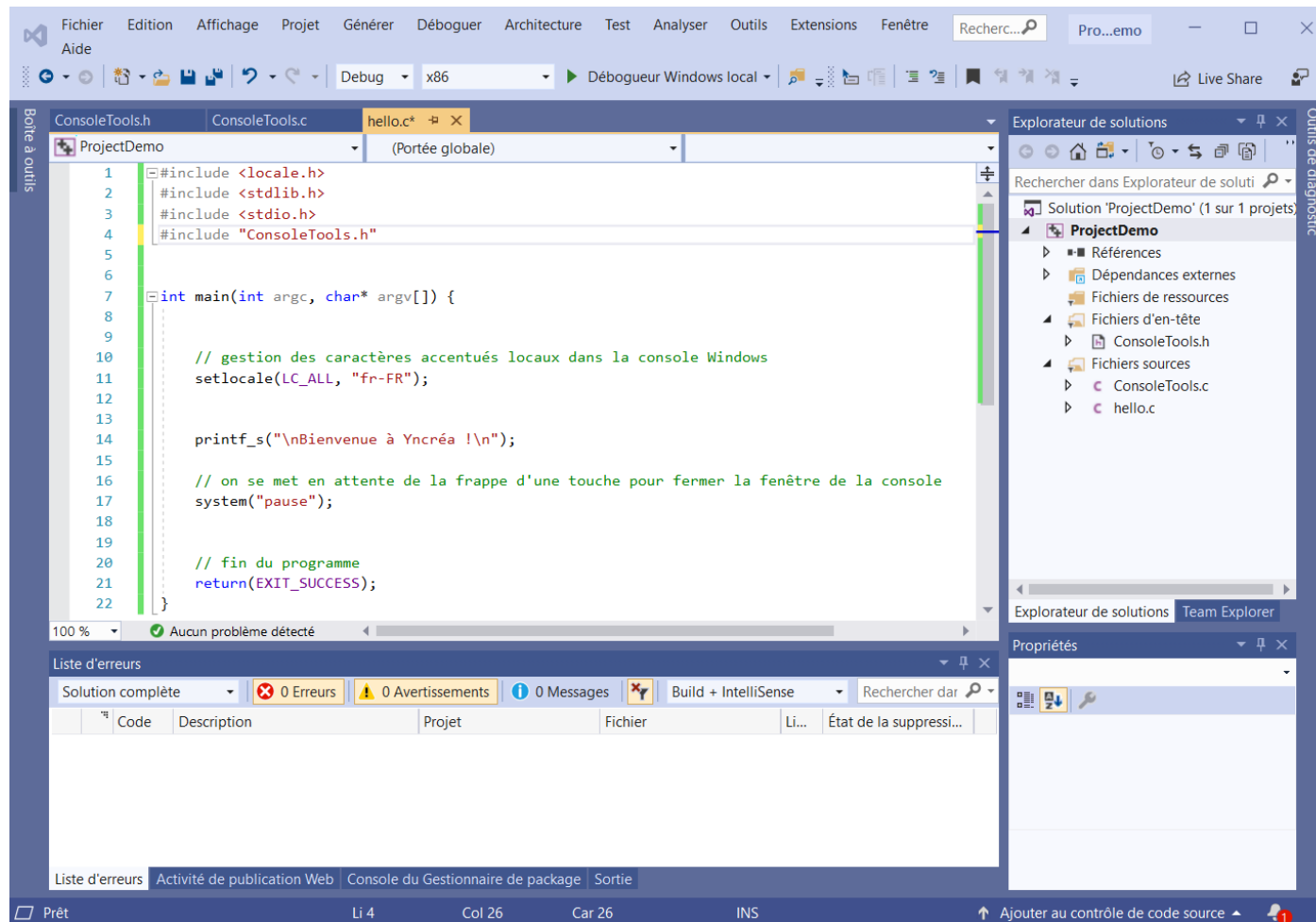
Épingler à Accès rapide Copier Coller Couper Copier le chemin d'accès Copier le raccourci Presse-papiers Déplacer vers Copier vers Supprimer Renommer Organiser Nouveau dossier Nouvel élément Accès rapide Propriétés Ouvrir Sélectionner tout Historique Modifier Inverser la sélection Sélectionner

← → ↶ ↷ > > Pascal Ricq > Source > Repos > ProjectDemo > ProjectDemo Rechercher dans : ProjectDemo

Nom	Modifié le	Type	Taille
Debug	04/09/2019 19:09	Dossier de fichiers	
<input checked="" type="checkbox"/> ConsoleTools.c	04/09/2019 18:57	Fichier source C	2 Ko
<input checked="" type="checkbox"/> ConsoleTools.h	04/09/2019 19:03	C/C++ Header	1 Ko
hello.c	04/09/2019 19:09	Fichier source C	1 Ko
ProjectDemo.vcxproj	04/09/2019 18:00	VC++ Project	8 Ko
ProjectDemo.vcxproj.filters	04/09/2019 18:00	VC++ Project Filte...	2 Ko
ProjectDemo.vcxproj.user	04/09/2019 16:27	Per-User Project O...	1 Ko

7 élément(s) | 2 éléments sélectionnés 2,91 Ko

On peut y ajouter d'autres fichiers et les ajouter au projets, puis inclure les nouveaux entêtes dans le code

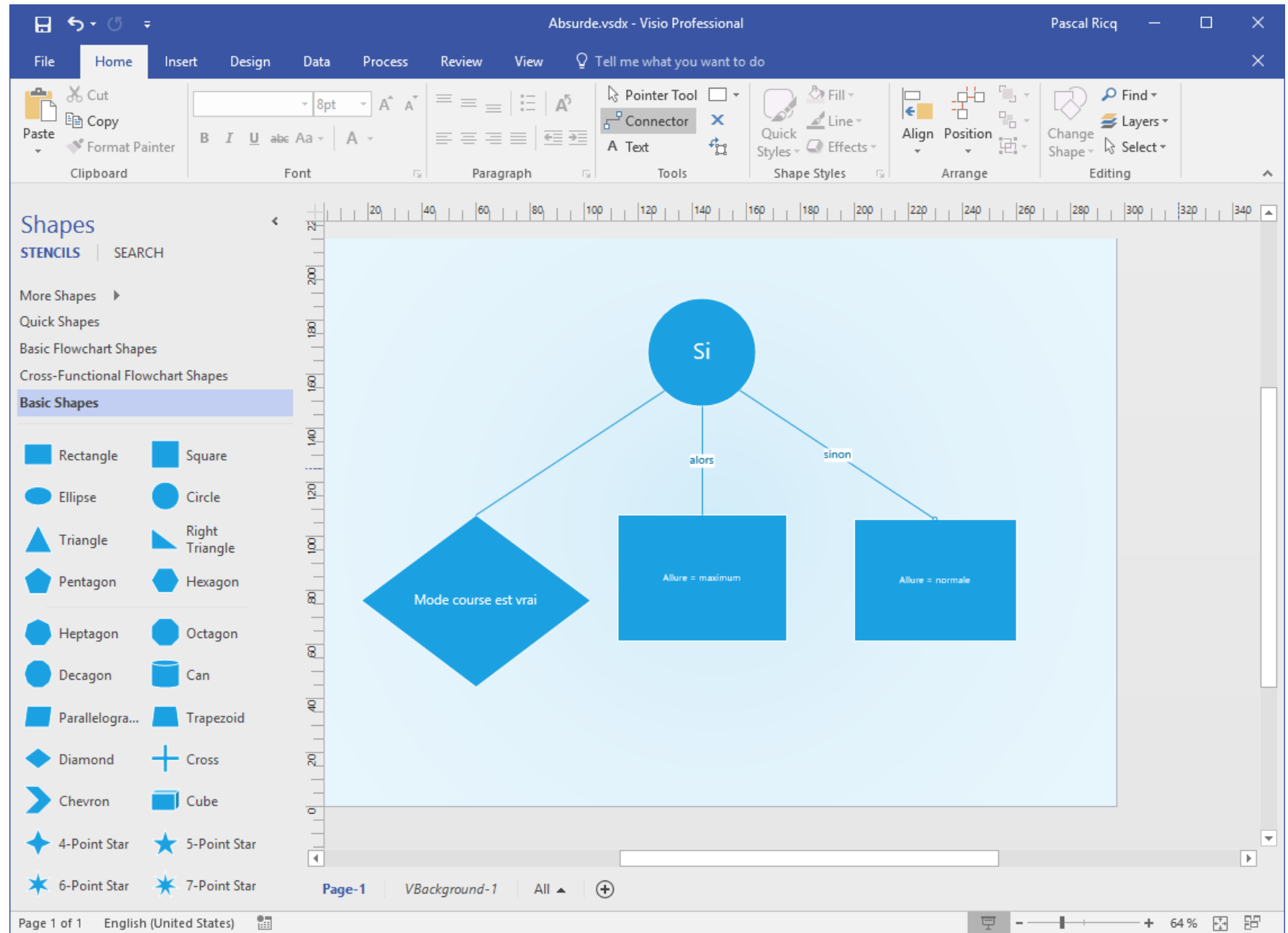




Et Aussi,

Visio

Installez Visio pour
dessiner des
organigrammes ou
des arbres
programmatiques.



Introduction à l'algorithmique

Analyser le problème

- C'est tout d'abord
 - Lire un énoncé
 - Faire l'interview d'un utilisateur afin de **recueillir ses besoins** et préciser avec lui ce qu'il attend du programme, les différents scénarios d'utilisation.
 - Comprendre le **périmètre du problème** et éliminer les informations parasites
- Ensuite, Il faut **identifier**
 - Les données en entrée
 - Les données en sortie
 - Les traitements à effectuer sur les données en entrée afin de produire les données attendues en sortie, tout en essayant de décomposer ces traitements en traitements simples.

Traduire les traitements en algorithmes

- L'objectif du cours d'algorithmique :
 - Il s'agit de découvrir les « briques » de base algorithmiques qui, assemblées entre elles vont permettre de décrire des traitements plus complets.
- A ce stade, il n'est pas encore question de programmation dans un langage de programmation, l'objectif est
 - utiliser des formalises simples (graphiques ou littéraux) sans être contraint par une syntaxe particulière. Ces algorithmes ne sont pas exécutables directement par la machine et devront passer par une phase de codage.



Traduire les traitements en algorithmes

Dans le cadre de la conception détaillées, différents formalismes textuels ou graphiques vont permettre de structurer le programme avec plus ou moins d'efficacité.

- Les traitements en codes machine font largement appelent à des « sauts » dans le code et l'**organigramme** classique reflète assez bien ce mécanisme en permettant lui aussi les branchements à des endroits quelconques du traitement, avec ou sans condition.
- Les **arbres programmatiques** sont plus difficiles à mettre en place mais offrent une meilleure structuration en éliminant les comportements hasardeux liés à des sauts non prévus. Les arbres et certains langages de programmation ne proposent pas la possibilité de faire des sauts. (cf instruction GOTO)



Traduire les traitements en algorithmes

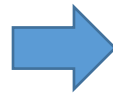
Les langages de programmation impératifs qui exécutent leurs instructions de manière séquentielle, afin de modifier l'état du programme (par exemple changer la valeur d'une somme) se prête bien à une conception algorithmique à l'aide d'organigramme ou d'arbre.

D'autres paradigmes de programmation, comme la programmation orientée objet ou événementielle, sont peu concernés par ces formalises et ont vraisemblablement contribué à leur inutilisation.

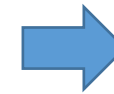
Exemple

On souhaite écrire un programme qui calcule la circonférence d'un cercle.

Données en entrée



Algorithme de traitement

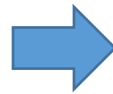


Données en sortie

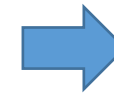
On souhaite écrire un programme qui calcule la circonférence d'un cercle.

L'utilisateur entre la valeur du rayon au clavier.

Données en entrée



Algorithme de traitement



Données en sortie

- la valeur du rayon en mètres
- la valeur de la constante π

On souhaite écrire un programme qui calcule la circonférence d'un cercle.

L'utilisateur entre la valeur du rayon au clavier.

Données en entrée



Algorithme de traitement



Données en sortie

- la valeur du rayon en mètres
- la valeur de la constante PI

application de la formule:
$$\text{Circonférence} = 2 \times \text{Rayon} \times \text{PI}$$

On souhaite écrire un programme qui calcule la circonférence d'un cercle.

Données en entrée



Algorithme de traitement



Données en sortie

- la valeur du rayon en mètres
- la valeur de la constante PI

application de la formule:
$$\text{Circonférence} = 2 \times \text{Rayon} \times \text{PI}$$

la valeur calculée de la
Circonférence du cercle

On souhaite écrire un programme qui calcule la circonférence d'un cercle. Le cercle à un rayon égal à 10 mètres.

Début

Faire

$\text{Circonférence} \leftarrow 10 \times 2 \times 3.1416$

Fait

Fin.



Exemple

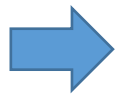
On souhaite écrire un programme qui calcule la circonférence d'un cercle. le cercle à un rayon égal à 10 mètres.

- Quels sont les limitations ?
- Quelles sont les améliorations possibles ?

- Quelles sont les limitations ?

- Le programme ne fonctionne que pour une valeur de rayon égale à 10
- L'utilisateur n'a pas la possibilité de choisir une nouvelle valeur sans réécrire un nouveau code
- On ne fait rien avec le résultat calculé

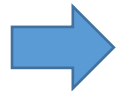
- Quelles sont les améliorations possibles ?



- Il faut rendre le **code réutilisable**
- Il faut gérer un dialogue avec l'utilisateur lorsque les données en entrée ou en sortie ne proviennent pas d'une autre partie du code

On souhaite écrire un programme qui calcule la circonférence d'un cercle. L'utilisateur entre au clavier la valeur du rayon.

Comment interagir avec l'utilisateur ? (Entrée / Sortie)



Afficher un message à l'écran (Sortie): Ecrire, Afficher
(Write, Print)

Obtenir une valeur au clavier (Entrée): Lire (Read)

On souhaite écrire un programme qui calcule la circonférence d'un cercle. L'utilisateur entre la valeur du rayon au clavier.

Comment rendre le code réutilisable :

- On utilise des variables qui vont permettre de généraliser le code.
- Il est utile à ce stade de préciser de quelle nature (type) sont les données (Entières, Décimales, ..) à stocker dans les variables
- Identifier les valeurs à définir en tant que constantes

Exemple 1:

Début

Constante : $\text{Pi} \leftarrow 3.1416$

Variable :

En Décimal : Rayon, Circonference

Partie déclarations

Faire

Afficher « Entrez une valeur de rayon »

Lire Rayon

Circonférence $\leftarrow \text{Rayon} \times 2 \times \text{Pi}$

Afficher « La circonférence d'un cercle de rayon » Rayon « est égale à » Circonference

Partie instructions

Fait

Fin.

Exemple 1:

Faire

1 - Afficher « Entrez une valeur de rayon »

2 - Lire Rayon

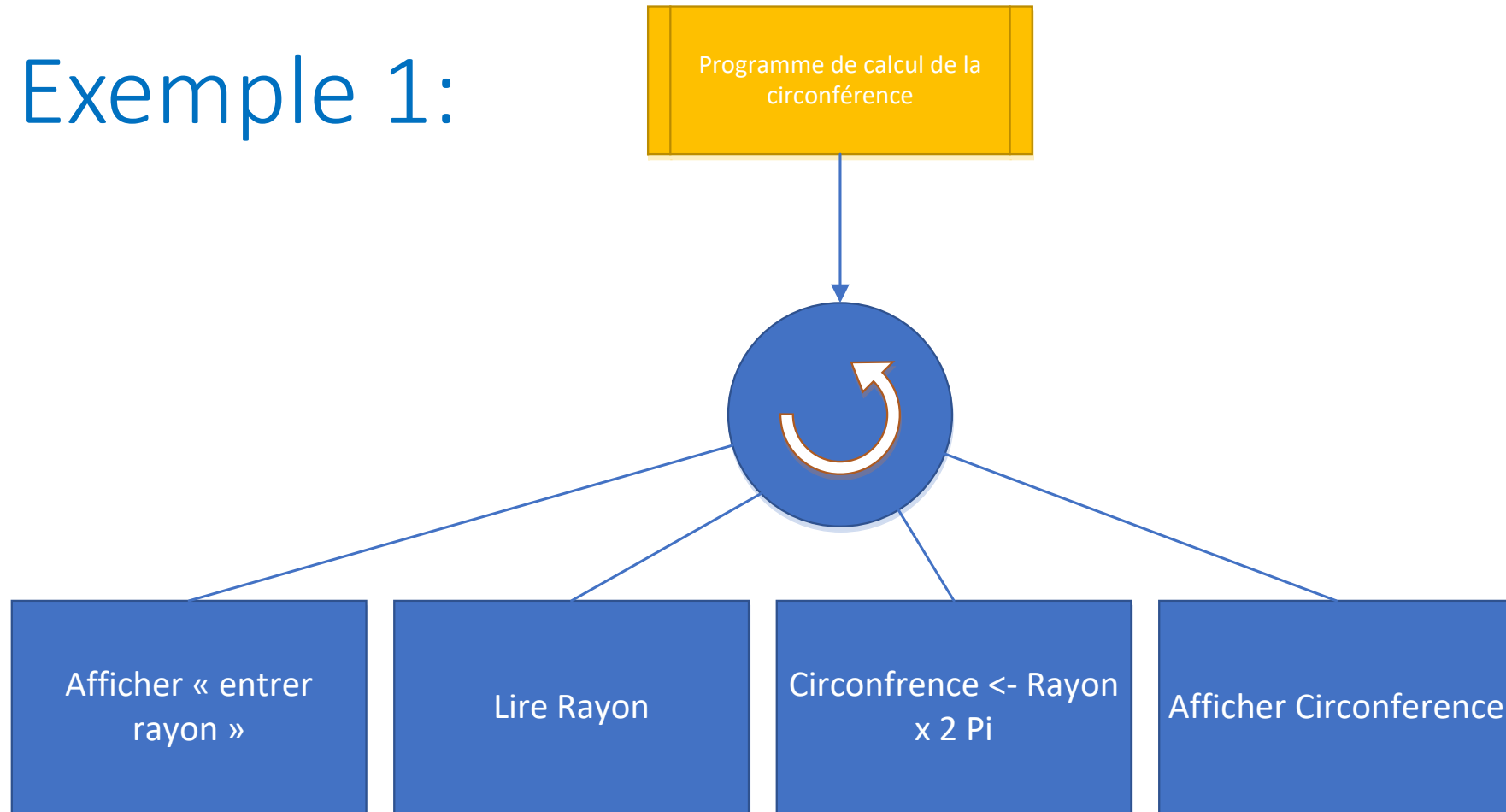
3 - Circonférence \leftarrow Rayon $\times 2 \times \text{Pi}$

4 - Afficher « La circonférence d'un cercle de rayon » Rayon « est égale à » Circonference

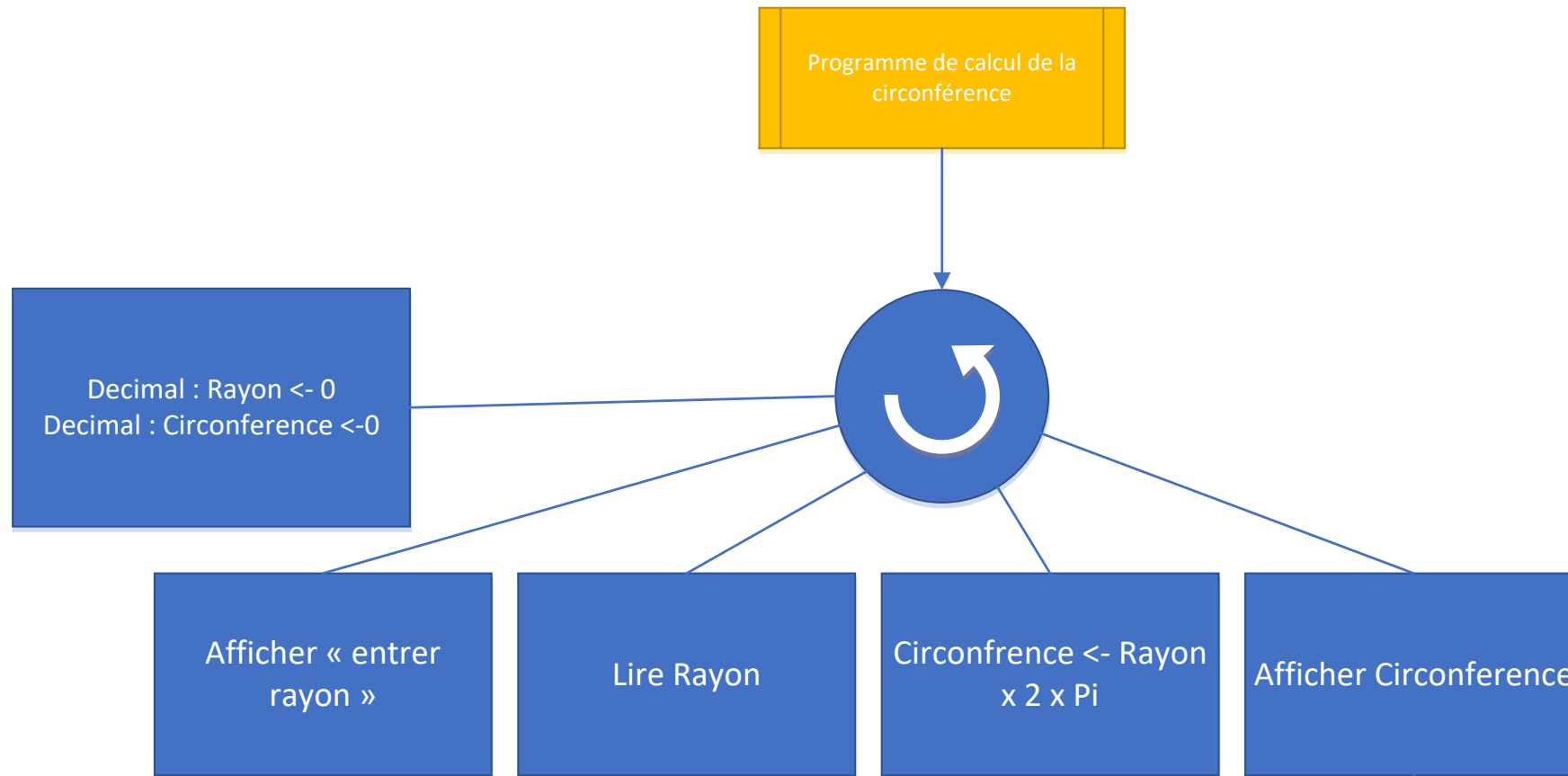
Fait

- Dans cet exemple, on réalise une **séquence** d'instructions regroupées dans un **bloc d'instructions** ici délimité par les mots clefs **Faire** et **Fait**.
- Toutes les instructions seront exécutées en commençant par la première jusqu'à la quatrième.

Exemple 1:



Représentation sous forme d'arbre d'une **séquence** d'instructions : La flèche indique l'ordre d'exécution.

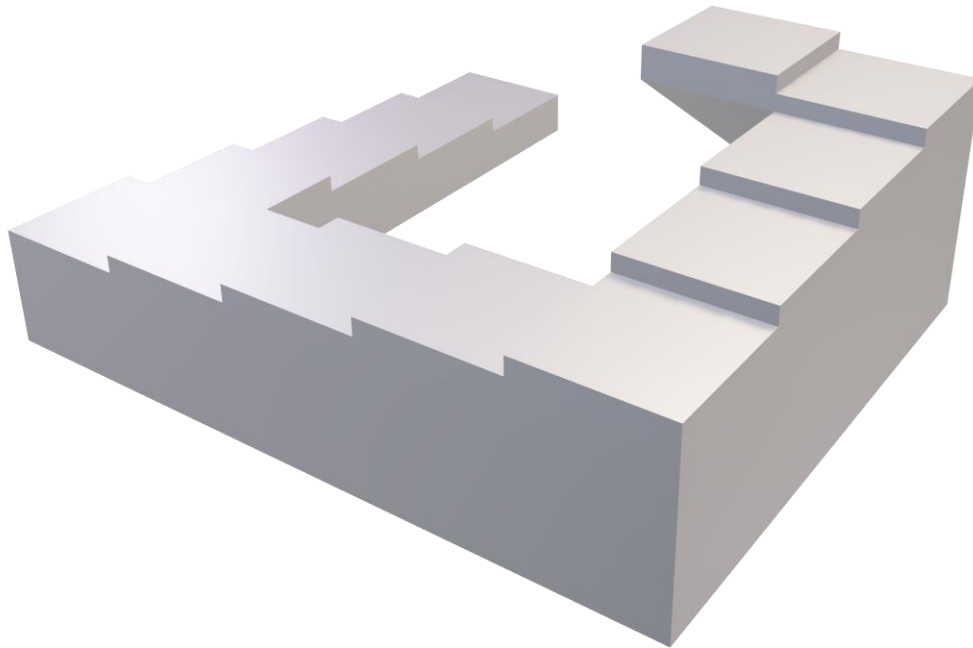


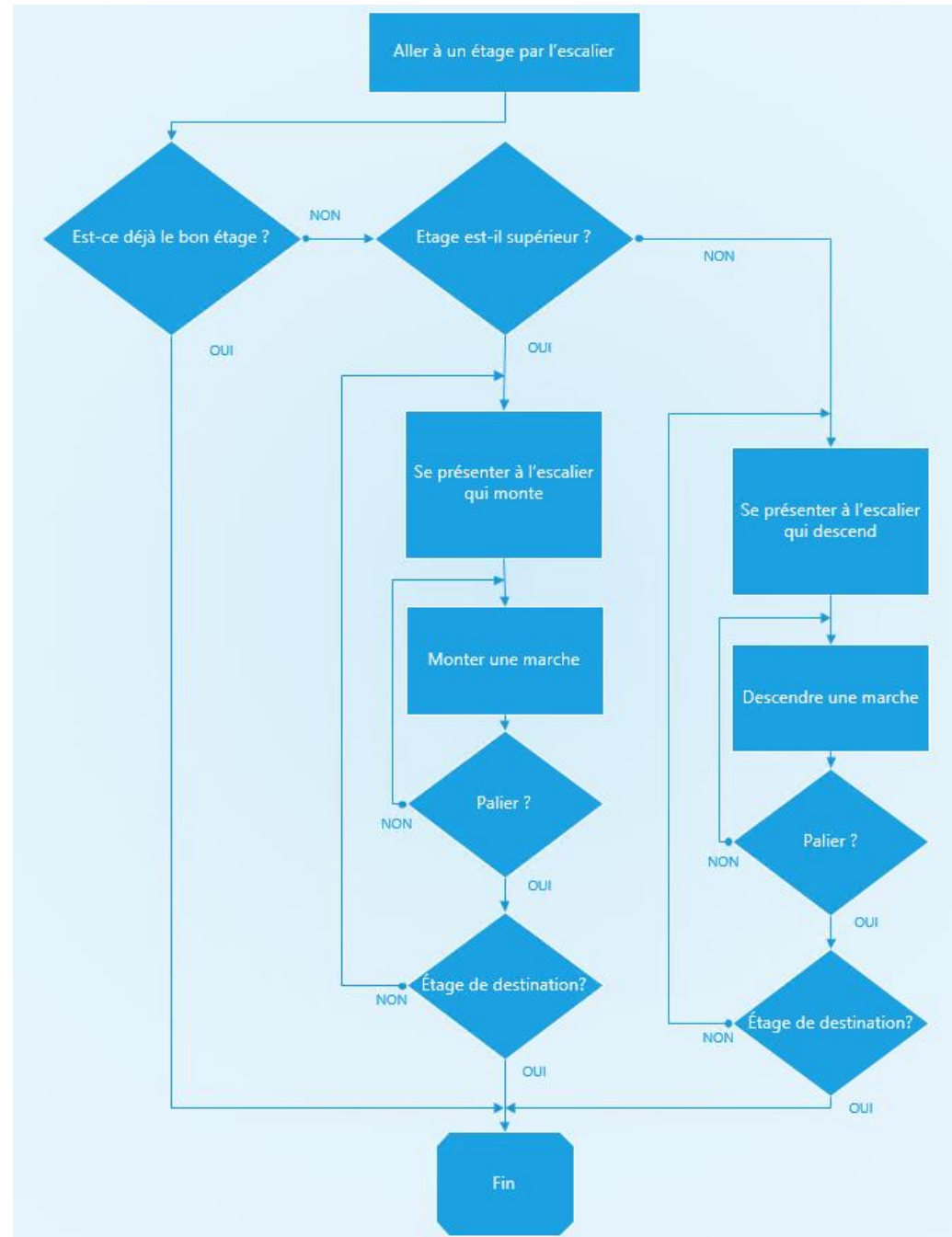
Il faut assigner une valeur aux variables avant de les faire intervenir dans une expression

- Les valeurs par défaut ne sont pas nécessairement attribuées ou connues.
- Certains langages vont refuser de compiler le programme lorsque que la variable n'a pas été assignée avant son utilisation, d'autres langages sont plus permissifs et l'erreur apparaîtra lors de l'exécution du programme.

Autre représentation d'un algorithme : l'organigramme

Aidez votre droïd à rejoindre un étage par l'escalier quand on lui dit par exemple « ok Droïd, rejoins-moi au 5^e » .





Proposez le **pseudo-code** et l'**arbre programmatique** du programme décrit par l'**organigramme** précédent

