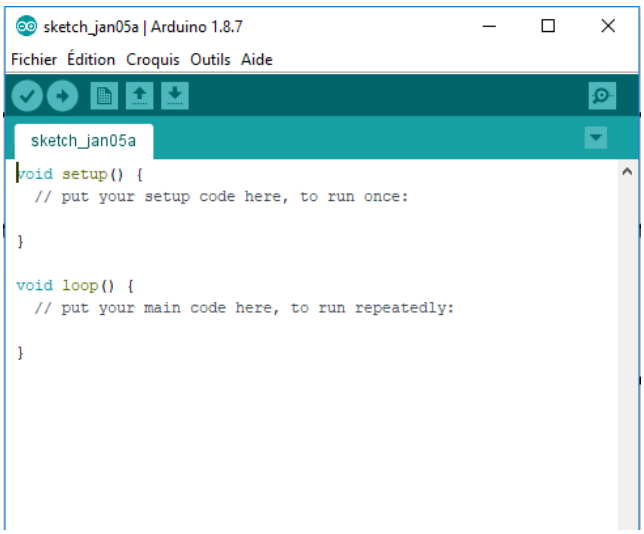
I) Structure d'un programme en C++ :

Quand tu lances le logiciel Arduino, le première fenêtre qui apparaît est la suivante :



Le texte inscrit dans la fenêtre est en ***langage C***. On distingue deux zones délimitées par des accolades :

- void setup() {}
- void loop() {}

La fonction ***setup()*** est une fonction qui ne s'exécute qu'une seule fois au démarrage du programme. C'est dans cette fonction qu'on déclare et initialise les variables ou qu'on définit la fonction de chaque broche (entrée ou sortie) par exemple.

La fonction ***loop()*** est la fonction principale de l'Arduino : une fois la fonction setup() exécutée, la fonction loop s'exécutera et se répétera à l'infini. Elle correspond au bloc ***Boucle*** d'Ardublock.

II) La fonction setup() :

Dans la fonction ***setup()***, on doit en premier lieu déclarer si les ports Dx seront utilisés comme des entrées ou des sorties. On utilise pour cela les instructions suivantes :

- pinMode (2, OUTPUT) ; → Déclare que D2 est une sortie sur laquelle on branchera un actionneur,
- pinMode (5, INPUT) ; → Déclare que D5 est une entrée sur laquelle on branchera un capteur.

Attention ! Il faut respecter la casse des caractères (majuscule ou minuscule) et ne pas oublier le point virgule à la fin des instructions.

III) La fonction loop() :

Dans la fonction ***loop()***, on peut utiliser des instructions qui seront équivalentes aux blocs employés dans Ardublock.

digitalWrite (2,HIGH) ;

digitalWrite (4,LOW) ;

delay(1000) ;

Mon premier programme :

Écris un programme en C qui permette de faire clignoter une diode. Téléverse-le vers l'Arduino en utilisant le bouton .

IV) Les instructions conditionnelles :

Il est bien sur possible de tester l'état d'une entrée, pour savoir par exemple si un bouton poussoir est enfoncé ou relâché.

```
if (digitalRead(2)==1)
  {
  }
```


```
if (digitalRead(2)==0)
  {
  }
```

Attention ! Il n'y a pas de point-virgule après un ***if***. Toutes les instructions qui doivent être exécutées seront placées entre les accolades.

```
if (digitalRead(2)==1)
  {
  }
  else
  {
  }
```

Exercice :

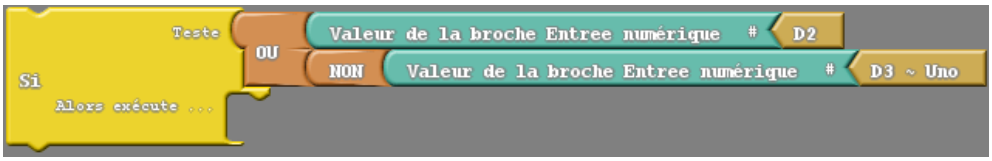
Une diode s'allume tant qu'un bouton poussoir est relâché, elle s'éteint sinon.

V) Les opérateurs logiques :

Un **ET** se traduit par **&&** en langage C.
Un **OU** se traduit par **||** (*AltGr + 6*).
Un **NON** se traduit par **!** .

Exemple :

```
if (digitalread(2)==1 || digitalRead(3)==0)
{}
```



Exercice :
Une diode s'allume si l'on appuie simultanément sur deux boutons poussoirs. Elle s'éteint sinon.

VI) Les entrées analogiques.

On a vu que sur un Arduino Uno, il y avait 4 entrées analogiques. Pour les tester, on va utiliser l'instruction suivante :

```
analogRead (0);
```

Pour pourvoir exploiter la valeur lue, on va utiliser les comparateurs suivant :

==	→	Égal à	<	→	Inférieur à
!=	→	Différent de	>=	→	Supérieur ou égal à
>	→	Supérieur à	<=	→	Inférieur ou égal à

Exemple :

```
if (analogRead(0) > 250)
{
}
```

Exercice :
Une diode s'allume si il fait sombre, elle s'éteint sinon.

Si tu as compris…
Avec ce que tu as appris, tu es maintenant capable de programmer en C++ l'éclairage automatique de la maison.