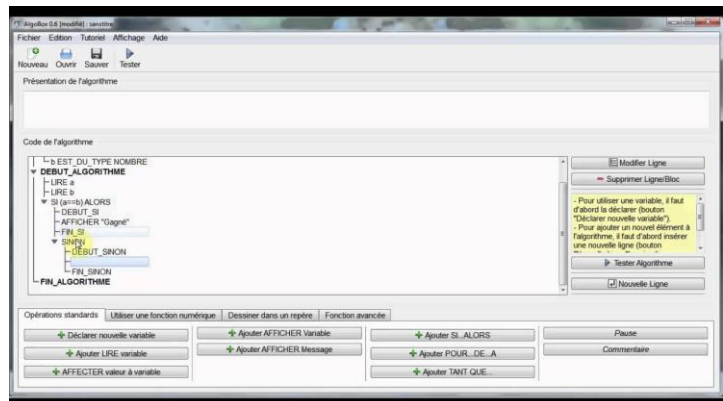
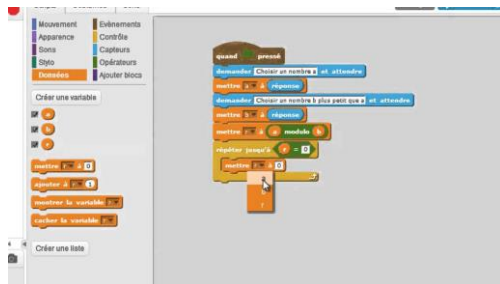


Cahier d'algorithmique – Programmation à l'aide de Scratch, de Algobox et du langage C



4^{ème} année

Table des matières

Histoire.....	3
Normalisation.....	4
Partie 1 : Comment utiliser ce cours	5
Partie 2 : Tes premières activités en C.....	6
Activité 1 : Ton premier programme en C : Dire Bonjour.....	6
Activité 2 : Affecter une valeur à une variable	7
Activité 3 : Récupérer une saisie.....	8
Activité 4 : Les opérations	9
Activité 7 : Le SINON	12
Activité 8 : Exercice sur le SINON.....	13
Activité 9 : Exercice sur le SINON SI	14
Activité 10 : Switch.....	15
Activité 11 : Gérer un menu avec un Switch	16
Activité 12 : La boucle While	17
Activité 13 : Le compteur	18
Activité 14 : La boucle Do While	19
Activité 15 : La boucle FOR.....	20
Activité 16 : Mettre en parallèle la programmation avec Scratch – le créateur d’algorithme Algobox et le langage C.....	21
Partie 3 : Calcul numérique.....	23
Activité 17 : Algorithme du prix à la caisse.....	23
Activité 18 : Algorithme de dépassement de budget	25
Activité 19 : Exercices dans Algobox et en C	27
Solutions.....	28
Bibliographie.....	31

Histoire

Le langage C a été inventé au cours de l'année 1972 dans les Laboratoires Bell. Il était développé en même temps que UNIX par Dennis Ritchie et Ken Thompson. Ken Thompson avait développé un prédécesseur de C, le langage B, qui est lui-même inspiré de BCPL. Dennis Ritchie a fait évoluer le langage B dans une nouvelle version suffisamment différente, en ajoutant notamment les types, pour qu'elle soit appelée C1.

Bien que C soit officiellement inspiré de B et de BCPL, on note une forte influence de PL/I (ou de PL360) ; on a pu dire que C était à Unix et au PDP-11 ce que PL/I fut pour la réécriture de Multics.

Par la suite, Brian Kernighan aida à populariser le langage C. Il procéda aussi à quelques modifications de dernière minute.

En 1978, Kernighan fut le principal auteur du livre *The C Programming Language* décrivant le langage enfin stabilisé ; Ritchie s'était occupé des appendices et des exemples avec Unix. On appelle aussi ce livre « le K&R », et l'on parle de C traditionnel ou de C K&R lorsqu'on se réfère au langage tel qu'il existait à cette époque.

Normalisation

En 1983, l'Institut national américain de normalisation (ANSI) a formé un comité de normalisation (X3J11) du langage qui a abouti en 1989 à la norme dite ANSI C ou C89 (formellement ANSI X3.159-1989). En 1990, cette norme a également été adoptée par l'Organisation internationale de normalisation (C90, C ISO, formellement ISO/CEI 9899:1990).

ANSI C est une évolution du C K&R qui reste extrêmement compatible. Elle reprend quelques idées de C++, notamment la notion de prototype et les qualificateurs de type².

Entre 1994 et 1996, le groupe de travail de l'ISO (ISO/CEI JTC1/SC22/WG14) a publié deux correctifs et un amendement à C90 : ISO/CEI 9899/COR1:1994 Technical Corrigendum 1, ISO/CEI 9899/AMD1:1995 Intégrité de C et ISO/CEI 9899/COR1:1996 Technical Corrigendum 2.

Ces changements assez modestes sont parfois appelés C89 avec amendement 1, ou C94 / C953,4. Trois fichiers d'entêtes ont été ajoutés, dont deux concernant les caractères larges et un autre définissant un certain nombre de macros en rapport avec la norme de caractères ISO 646.

En 1999, une nouvelle évolution du langage est normalisée par l'ISO : C99 (formellement ISO/CEI 9899:1999). Les nouveautés portent notamment sur les tableaux de taille variable, les pointeurs restreints, les nombres complexes, les littéraux composés, les déclarations mélangées avec les instructions, les fonctions inline, le support avancé des nombres flottants, et la syntaxe de commentaire de C++.

La bibliothèque standard du C99 a été enrichie de six fichiers d'en-tête depuis la précédente norme.

En 2011, l'ISO ratifie une nouvelle version du standard⁵ : C11, formellement ISO/CEI 9899:2011. Cette évolution introduit notamment le support de la programmation multi-thread, les expressions à type générique, et un meilleur support d'Unicode.

Partie 1 : Comment utiliser ce cours

- Lancer le programme CodeBlock
- Cliquer sur File - New - Empty File
- Il ne vous reste plus qu'à programmer

- Les explications à donner pour chaque exercice sont recherchées dans le cours : Langage C avec CODEBLOCK présent sur le site du professeur [PROINFOBIK .JIMDO.COM](http://PROINFOBIK.JIMDO.COM)

- Chaque exercice est nommé : Act1, Act2, Act3, etc

Partie 2 : Tes premières activités en C

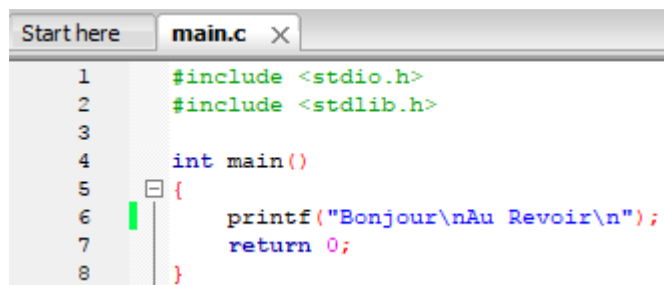
Activité 1 : Ton premier programme en C : Dire Bonjour

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("Hello world!\n");
7      return 0;
8  }
```

Explications :

- #include <stdio.h> :
- #include <stdlib.h> :
- Int main()
- Printf
- \n
- Return 0 ;

Modifie le programme précédent et enregistre-le avec le même nom :



```
Start here  main.c  X
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("Bonjour\nAu Revoir\n");
7      return 0;
8  }
```

Activité 2 : Affecter une valeur à une variable

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int age;
7      age = 18;
8
9      return 0;
10 }
```

Explication :

□ int :

Activité 3 : Récupérer une saisie

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int age = 0; // On initialise en premier la variable à 0*/
7      printf("Quel age avez-vous ? ");
8      scanf("%d", &age); //On demande d'entrer ton age avec scanf
9      printf("Ah ! Vous avez donc %d ans !\n\n", age);
10     return 0;
11 }
```

Que fait ce programme, explique :

Explications :

- scanf :
- %d :

Activité 4 : Les opérations

Act 4.1 : Addition

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int resultat = 0;
7      resultat = 5 + 3;
8      printf("5 + 3 = %d", resultat);
9      return 0;
10 }
```

Act 4.2 : Division

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      double resultat = 0;
7      resultat = 5.0 / 2.0;
8      printf("5 / 2 = %f", resultat);
9      return 0;
10 }
```

Act 4.3 : Modulo

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int resultat = 0;
7      resultat = 14 % 3;
8      printf("14 modulo 3 = %d", resultat);
9      return 0;
10 }
```

Activité 5 : Des calculs entre variable

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int resultat = 0, nombre1 = 0, nombre2 = 0;
7      /*On demande l'introduction du nombre 1 et 2 à l'utilisateur*/
8      printf("Entrez le nombre 1 :\n");
9      scanf("%d", &nombre1);
10     printf("Entrez le nombre 2 :\n");
11     scanf("%d", &nombre2);
12     //On réalise un calcul
13     resultat = nombre1 + nombre2;
14     //Et on affiche l'addition à l'écran
15     printf("Voici le resultat de l'addition : %d + %d = %d\n", nombre1, nombre2, resultat);
16
17     return 0;
18 }
```

Que fait ce programme, explique :

Activité 6 : Les conditions if

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      long x = 5;
7      long y = 3;
8
9      /* si x est supérieur à y */
10     if (x > y)
11         printf("X est supérieur a Y");
12
13     return 0;
```

Que fait ce programme, explique :

Explications :

Long :

if :

Modifie ton travail précédent :

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      long x = 2;
7      long y = 7;
8
9      /* si x est supérieur à y */
10     if (x > y)
11         printf("X est supérieur a Y\n");
12     if (x < y)
13         printf("X est inférieur a Y\n");
14
15     return 0;
16 }
```

Modifie ton travail précédent :

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      long x = 2;
7      long y = 7;
8
9      /* si x est supérieur à y */
10     if (x > y)
11         printf("X est supérieur a Y\n");
12     if (!(x > y))
13         printf("X est inférieur a Y\n");
14
15     return 0;
16 }
```

Explication :

!:

Activité 7 : Le SINON

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      long x = 2;
7      long y = 7;
8
9      /* si x est supérieur à y */
10     if (x > y)
11     {
12         printf("X est supérieur a Y\n");
13     }
14     else
15     {
16         printf("X est inférieur a Y\n");
17     }
18     return 0;
19 }
```

Que fait ce programme, explique :

Explication :

else :

Activité 8 : Exercice sur le SINON

On demande l'âge, si l'âge est supérieur ou égal à 18 alors affiche « Vous êtes majeur » sinon affiche « Tu es mineur ».

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      long age = 0;
7
8      printf("Quel est votre age ? ");
9      scanf("%ld", &age);
10
11     /*Si age = 18 ou (ou = comme opérateur ||) si age est > à 18 */
12     if (age == 18 || age > 18)
13     {
14         printf("Vous etes majeur");
15     }
16     else
17     {
18         printf("tu es mineur");
19     }
20     return 0;
21 }
```

Explications :

□ == :

□ ||

Activité 9 : Exercice sur le SINON SI

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      long age = 0;
7
8      printf("Quel est votre age ? ");
9      scanf("%ld", &age);
10
11     /*Si age = 18 ou (ou = comme opérateur ||) si age est > à 18 */
12     if (age < 18)
13     {
14         printf("Vous etes mineur");
15     }
16     else if (age ==18)
17     {
18         printf("Tu es tout juste majeur");
19     }
20     /*si age est > 18 et (&&) age est < 100*/
21     else if (age >18 && age <100)
22     {
23         printf("Vous etes majeur");
24     }
25
26     else
27     {
28         printf("Tu dois etre tres age ?");
29     }
30     return 0;
31 }
```

Que fait ce programme par rapport au précédent, explique :

Explication :

else if :

Activité 10 : Switch

Alors, pour éviter d'avoir à faire des répétitions dans les tests, ils ont inventé une autre structure que le if... else. Cette structure particulière s'appelle switch.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int age = 0;
7
8      printf("Quel age avez-vous ? ");
9      scanf("%d", &age);
10
11     switch (age)
12     {
13         case 2:printf("Salut bebe !");
14             break;
15
16         case 6:printf("Salut gamin !");
17             break;
18
19         case 12:printf("Salut jeune !");
20             break;
21
22         case 16:printf("Salut ado !");
23             break;
24
25         case 18:printf("Salut adulte !");
26             break;
27
28         case 60:printf("Salut papy !");
29             break;
30
31         default:printf("Je n'ai aucune phrase de prete pour ton age ");
32             break;
33     }
34 }
```

Explication :

- Case :
- Break :
- Switch :

Activité 11 : Gérer un menu avec un Switch

En console, pour faire un menu, on fait des printf qui affichent les différentes options possibles. Chaque option est numérotée, et l'utilisateur doit entrer le numéro du menu qui l'intéresse.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      int choixMenu;
7
8      printf("=== Menu ===\n\n");
9      printf("1. Royal Cheese\n");
10     printf("2. Mc Deluxe\n");
11     printf("3. Mc Bacon\n");
12     printf("4. Big Mac\n");
13     printf("\nVotre choix ? ");
14     scanf("%d", &choixMenu);
15     printf("\n");
16
17     switch (choixMenu)
18     {
19         case 1:
20             printf("Vous avez choisi le Royal Cheese. Bon choix !");
21             break;
22
23         case 2:
24             printf("Vous avez choisi le Mc Deluxe. Berk, trop de sauce...");
25             break;
26
27         case 3:
28             printf("Vous avez choisi le Mc Bacon. Bon, ca passe encore ca ;o)");
29             break;
30
31         case 4:
32             printf("Vous avez choisi le Big Mac. Vous devez avoir tres faim !");
33             break;
34
35         default:
36             printf("Vous n'avez pas rentre un nombre correct. Vous ne mangerez rien du tout !");
37             break;
38     }
39     printf("\n\n");
40     return 0;
41 }
```


Activité 12 : La boucle While

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int nombreEntre = 0;
7      while (nombreEntre != 47)
8      {
9          printf("Tapez le nombre 47 ! ");
10         scanf("%d", &nombreEntre);
11     }
12 }
```

Que fait ce programme, explique :

Explication :

While :

Activité 13 : Le compteur

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int compteur = 0;
7      while (compteur < 10)
8      {
9          printf("Salut les Zeros !\n");
10         compteur++;
11     }
12 }
```

Que fait ce programme, explique :

Explication :

++ :

Activité 14 : La boucle Do While

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int compteur = 0;
7      do
8      {
9          printf("Salut les informaticiens !\n");
10         compteur++;
11     }
12     while (compteur < 10);
13 }
```

Différence avec la boucle While, explique :

Activité 15 : La boucle FOR

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int compteur;
7      for (compteur = 0 ; compteur < 10 ; compteur++)
8      {
9          printf("Salut les Pingouins !\n");
10     }
11 }
```

Différence avec la boucle Do ... While, explique :

Activité 16 : Mettre en parallèle la programmation avec Scratch - le créateur d'algorithme Algobox et le langage C

Objectifs

Décoder et comprendre un algorithme. Transformer l'algorithme en programme dans Algobox.

Programmer un algorithme, c'est le traduire dans un langage compréhensible par un logiciel donné.

Algorithme à transformer dans Algobox.

A l'aide de deux variables introduites tu vas devoir créer un algorithme permettant le calcul du périmètre et de l'aire d'une surface.

A toi de jouer

Voici l'algorithme en Scratch que tu devrais programmer, transforme-le ensuite dans Algobox.

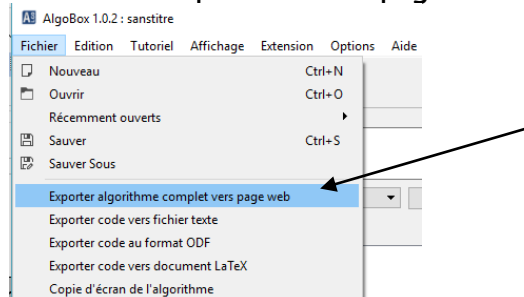
Énoncé :



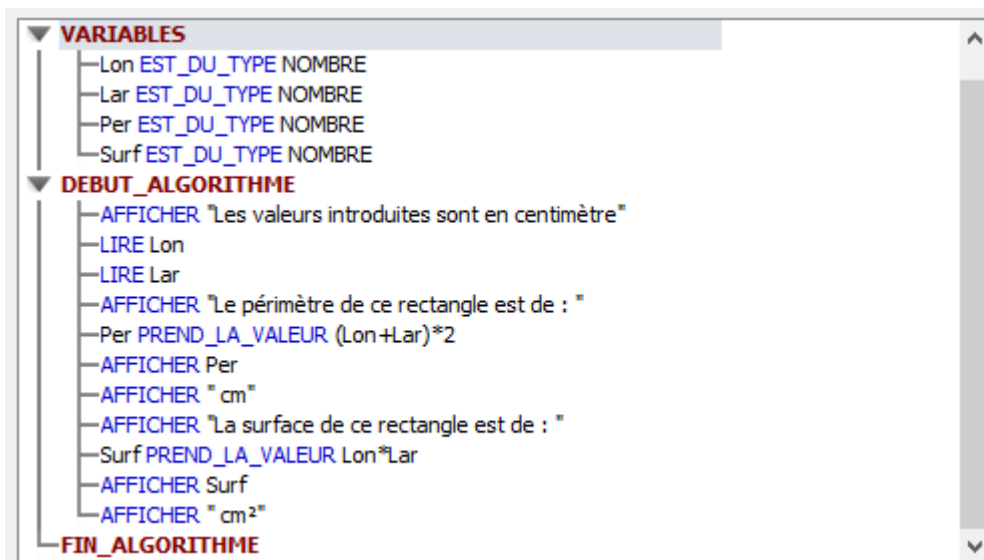
Ajoutez une ligne à votre programme en Scratch afin qu'il calcule aussi la surface du rectangle.

Transformez maintenant cet algorithme dans Algobox.

Enregistrez votre projet une fois terminé en HTML et n'oubliez pas de réaliser une référence à partir de votre page Web vers cet algorithme.



Solution dans Algobox :



Transforme maintenant ton algorithme en langage C à l'aide de CodeBlocks

Voici la solution que tu peux consulter si tu as des soucis.

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int L =0;
    int l =0;
    int P =0;
    int A =0;

    printf("Longueur : ");
    scanf("%d", &L);
    printf("Largeur : ");
    scanf("%d", &l);

    P=2*L+2*l;
    A=L*l;

    printf("Perimetre = %d cm| Aire = %d cm2", P , A);

    system("pause");

    return 0;
}
  
```

Partie 3 : Calcul numérique

Activité 17 : Algorithme du prix à la caisse

Objectif

Réaliser un calcul numérique à l'aide de variables

Ce que doit faire l'algorithme

Dans un magasin, chaque celf USB coûte 9,90 € et chaque DVD coûte 0,60 €. L'algorithme calcule le prix à la caisse en fonction du nombre de clefs et de DVD achetés.

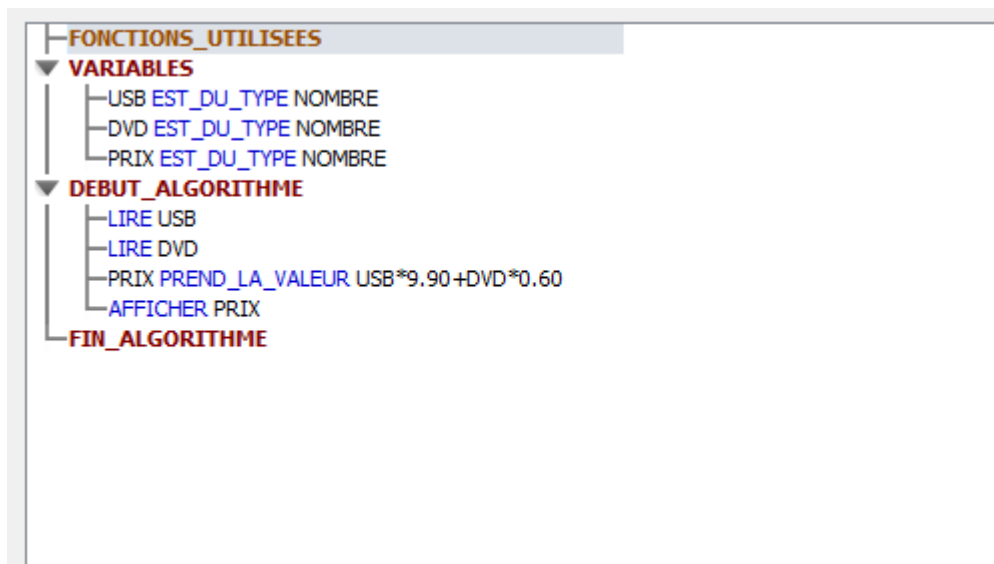
Les variables sont : USB, DVD, PRIX

A toi de jouer

Voici l'algorithme en Scratch. Essaie de le transformer dans Algobox.



Enregistre (Fichier-Exportation Page Web) ton projet Algobox en HTML.
La solution est à la page suivante.



Transforme maintenant ton algorithme en langage C à l'aide de CodeBlocks

Voici la solution que tu peux consulter si tu as des soucis.

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int USB = 0;
    int DVD = 0;
    int tot = 0;

    printf("Nombre de cle USB a 9.90 euro : ");
    scanf("%d", &USB);
    printf("Nombre de DVD a 0.60 euro : ");
    scanf("%d", &DVD);

    tot= USB*9.9+DVD*0.6;

    printf("Vous avez achete pour un total de : %d euro", tot);

    system("pause");

    return 0;
}

```


Activité 18 : Algorithme de dépassement de budget

Objectif

Prévenir quand un budget est dépassé à l'aide d'une comparaison

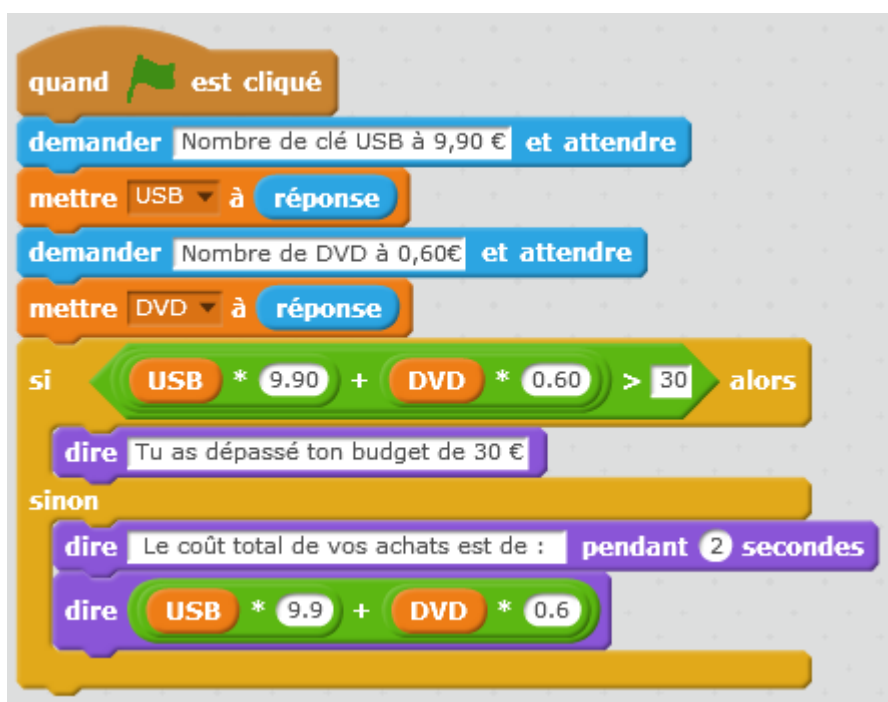
Ce que doit faire l'algorithme

Tu disposes d'un budget de 30 €. Tu veux acheter des clef USB de 9,90 € et des DVD de 0,6. L'algorithme doit te prévenir quand le budget est dépassé.

Les variables sont : USB, DVD et PRIX.

A toi de jouer

Voici l'algorithme en Scratch. Réalise-le et ensuite essaie de le transformer dans Algobox.



Enregistre ton projet Algobox en HTML.

La solution est à la page suivante.



Transforme maintenant ton algorithme en langage C à l'aide de CodeBlocks

Voici la solution que tu peux consulter si tu as des soucis.

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int USB = 0;
    int DVD = 0;
    int tot = 0;

    printf("Nombre de cle USB a 9.90 euro : ");
    scanf("%d", &USB);
    printf("Nombre de DVD a 0.60 euro : ");
    scanf("%d", &DVD);

    tot= USB*9.9+DVD*0.6;

    if (tot > 30)
    {
        printf("Vous avez depassez votre budget de 30 euro \n");
        printf("Vous devez payez %d euro \n", tot);
    }
    else
    {
        printf("Vous devez payez %d euro \n", tot);
    }
    system("pause");
    return 0;
}
  
```

Activité 19 : Exercices dans Algobox et en C

4.1 : 1^{er} algorithme à trouver. Ce que doit faire l'algorithme :

On considère deux variables A et B qui ont respectivement pour valeurs 5 et 7. On souhaite échanger à l'aide d'un algorithme les valeurs de A et de B. Affiche les nouvelles valeurs des deux variables.

4.2 : 2^{ème} algorithme à trouver. Ce que doit faire l'algorithme :

Pour ce trimestre, Suzanne a obtenu trois notes en mathématiques, désignée par les variables A, B et C. Crée un algorithme permettant à Suzanne de connaître sa moyenne du trimestre. Affiche la moyenne. Les cotations sont introduites par le professeur.

4.3 : 3^{ème} algorithme à trouver. Ce que doit faire l'algorithme :

Un commerçant s'apprête à modifier ses étiquettes. Deux produits seront en promotion. La valeur respective de chacun d'eux est de 33 € et 48 €. Sur le premier il va réaliser une remise de 15 % et sur le second la remise sera de 22 %.

Réalise un algorithme afin que le prix de chacun de ces produits soit une variable A et B. Affiche les nouveaux prix.

4.4 : 4^{ème} algorithme à trouver. Ce que doit faire l'algorithme :

Voici une recette de cuisine d'un gratin de pommes pour 4 personnes. Ingrédients : 1kg de pommes de terre, 60 g de fromage râpé, 25 cl de crème fraîche. Calcule à l'aide d'un algorithme la quantité de pommes de terre à acheter si 6 personnes viennent à son souper. Affiche la valeur. Même calcul dans le même algorithme pour la quantité de crème fraîche.

4.5 : 5^{ème} algorithme à trouver. Ce que doit faire l'algorithme :

On rappelle la formule : $D = V \times T$ où V est la vitesse moyenne (en km/h), D la distance parcourue (en km) et T la durée du parcours (en h). Aïda doit parcourir 572 km. Calcule la durée (en heures) de son parcours à l'aide d'un algorithme si sa vitesse moyenne est de 110 km/h. Tu peux créer cet algorithme avec ces valeurs mais par après, transforme-le afin d'utiliser des variables comme ci-dessus (D, V et T).

Solutions

1^{er} algorithme :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int A=5;
    int B=7;

    A=7;
    B=5;

    printf("A=%d B=%d",A ,B);

    system("pause");

    return 0;
}
```

2^{ème} algorithme :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int A=0;
    int B=0;
    int C=0;
    int moy=0;

    printf("Que vaut la quote 1 ? ");
    scanf("%d", &A);
    printf("Que vaut la quote 2 ? ");
    scanf("%d", &B);
    printf("Que vaut la quote 3 ? ");
    scanf("%d", &C);

    moy= (A+B+C)/3;

    printf("la moyenne de Suzanne est de %d      ", moy);

    system("pause");

    return 0;
}
```

3^{ème} algorithme :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int A=0;
    int B=0;

    A= 33*15/100;
    B= 48*22/100;

    printf("les nouveaux prix sont %d et %d\n", A, B);

    system("pause");

    return 0;
}
```

4^{ème} algorithme :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int A=1000;
    int B=60;
    int C=25;

    int D=0;
    int E=0;

    D=A/4*6;
    E=C/4*6;

    printf("La quantité de pommes de terre pour 6 personnes sera de %d gr \n\n", D);
    printf("La quantité de crème fraîche pour 6 personnes sera de %d cl \n\n", E);

    system("pause");

    return 0;
}
```

5^{ème} algorithme :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int D=570;
    int V=110;
    int T=0;

    T=D/V;

    printf("la duree du trajet est de %d heures      ", T);

    system("pause");

    return 0;
}
```

Bibliographie

- Tous les algorithmes : Programmation Algobox - Ellipses
- Cahier d'algorithme et de programmation : Initiation au codage avec Scratch - Delagrave
- Wikipédia