بسم الله الرحمن الرحيم

اسأل الله ان اجدكم في احسن حال و الصحة, انتم و ذويكم

السلام عليكم و رحمه الله و بركاته

Salam, j’ espère que vous allez bien ainsi que votre famille

La récursivité

Rappel et exercices

- ✓ Principe
- ✓ Implémentation
- ✓ Exemple
- ✓ Finalité
- ✓ Illustration d'une solution récursive
- ✓ Exercices 1-2-3 série 2
- ➢ développement des solutions des exercices

développement des solutions des exercices

EX2 : La suite de Fibonacci est définie comme suit :

$U_0=0$;

$U_1=1$;

$U_n=U_{n-1}+U_{n-2}$ pour $n>=2$.

Ecrire un sous-programme récursif qui calcule cette fonction.

- Avant de développer la solution itérative et récursive, vous pouvez remarquer que parfois le problème se défini lui-même, en forme récursive, ce qui suggère qu'il est plus judicieux d' adapter une solution récursive mieux qu'une solution itérative .
- Nous rappelons qu'il y a des problèmes dont la solution intuitive est récursive et qu'il est particulièrement difficile de trouver son équivalente itérative.

Solution itérative de Ex 2

$U_0=0$;
$U_1=1$;
$U_n=U_{n-1}+U_{n-2}$ pour $n>=2$

- Vous remarquez que U_n est composée des deux termes précédents U_{n-1} et U_{n-2}
- Prenons un exemple pour n = 5
- n = 5 $U_5 = U_4 + U_3$
- On ne peut appliquer la formule directement car nous ignorons les valeurs de U_4 et de U_3
- De plus, dans la solution itérative la fonction ne doit pas s'appeler
- Donc nous devons procéder aux calculs du premier terme U_0 jusqu'au terme U_5

U0 = 0
U1 = 1
U2 = U1 + U0 = 1 + 0 = 1
U3 = U2 + U1 = 1 + 1 = 2
U4 = U3 + U2 = 2 + 1 = 3
U5 = U4 + U3 = 3 + 2 = 5

Solution itérative de Ex 2

U0 = 0
U1 = 1
U2 = U1 + U0 = 1 + 0 = 1
U3 = U2 + U1 = 1 + 1 = 2
U4 = U3 + U2 = 2 + 1 = 3
U5 = U4 + U3 = 3 + 2 = 5

-En observant ces différentes opérations, nous constatons pour n = 5, nous devons avoir 6 termes : $U_0, U_1, U_2, U_3, U_4, U_5$ $\longrightarrow$ donc intuitivement 6 variables

- Et pour n = 7 $\longrightarrow$ nous aurons besoin de 8 variables

- Et donc pour $\forall$ n $\longrightarrow$ nous aurons besoin de (n+1) variables

Mais

Les langages de programmation, en général, ne permettent pas de définir le nombre de variables à déclarer ou de les déclarer lors de l'exécution.

Solution itérative de Ex 2

<u>Mais</u>

Les langages de programmation, en général, ne permettent pas de définir le nombre de variables à déclarer ou de les déclarer lors de l'exécution.

- En effet, il faut comprendre qu'on général, notre programme passe par deux phases avant d' être exécuter

- Une **première phase** pour chercher les erreurs et entre autre transformer notre programme au format binaire *.

- Une **deuxième phase** qui est l'exécution propre du programme au format binaire.

Solution itérative de Ex 2

- La déclaration des variables, se fait dans la **première phase** → donc, nous n'avons pas un moyen de connaitre le nombre de variables nécessaires pour calculer U_n → Pourquoi ?

Parce que la valeur de n ne sera donnée qu'en **deuxième phase**, lors de l' exécution.

*Rappel

Rappelons que l'ordinateur ne manipule que des données binaires.

Solution itérative de Ex 2

Alors , il faut dés le départ , décider du nombre de variables à utiliser $\forall n$

$$U0 = 0$$
$$U1 = 1$$
$$U2 = U1 + U0 = 1 + 0 = 1$$
$$U3 = U2 + U1 = 1 + 1 = 2$$
$$U4 = U3 + U2 = 2 + 1 = 3$$
$$U5 = U4 + U3 = 3 + 2 = 5$$

- Nous pouvons remarquer qu'à partir de n = 2, $\forall U_n$ est calculé de la même manière $\longrightarrow$ On peut les reformuler par $U_n = a + b$

Tel que $a = U_{n-1}$ et $b = U_{n-2}$

- chaque terme dépend des deux termes précédents uniquement

- Nous n'avons pas besoin de sauvegarder tout les termes calculés sauf les deux derniers

Solution itérative de Ex 2

$U0 = 0$
$U1 = 1$
$U2 = U1 + U0 = 1 + 0 = 1$
$U3 = U2 + U1 = 1 + 1 = 2$
$U4 = U3 + U2 = 2 + 1 = 3$
$U5 = U4 + U3 = 3 + 2 = 5$

- On transforme les formules précédentes, en prenant en compte les observations précédentes

$a = 0$
$b = 1$

$c = a + b = 1 + 0 = 1$	$b \leftarrow a$	$a \leftarrow c$	$b = 1$	$a = 1$
$c = a + b = 1 + 1 = 2$	$b \leftarrow a$	$a \leftarrow c$	$b = 1$	$a = 2$
$c = a + b = 2 + 1 = 3$	$b \leftarrow a$	$a \leftarrow c$	$b = 2$	$a = 3$
$c = a + b = 3 + 2 = 5$	$b \leftarrow a$	$a \leftarrow c$	$b = 2$	$a = 3$

Nous remarquons qu'on a réduit le nombre de variables à 3 $\forall$ n

Solution itérative de Ex 2

Solution itérative

```
Int Fibiterative (int n)
{
    int f, f0, f1, i ;
    f0 = 0;                    // f0 correspond à b  dans l' explication
    f1 = 1;                    // f1 correspond à a  dans l' explication
    for (i = 2;  i<= n;  i++)
     {
        f = f0 + f1 ;          // f  correspond à c  dans l' explication
        f0 = f1;
        f1 = f;
      }
      return f;
}
```

-Il nous ai demandé de calculer le terme $U_n \longrightarrow$ donc nous avons défini une fonction

- Tout les nombres manipulés sont des entiers $\longrightarrow$ donc la fonction retourne un entier.

Solution récursive de Ex 2

$U_0=0$;

$U_1=1$;

$U_n=U_{n-1}+U_{n-2}$ pour $n>=2$

- Comme mentionné précédemment, le problème est défini en forme récursive, ce qui suggère que la solution récursive est plus intuitive que sa version itérative .
- Nous utilisons la formule de U_n telle que donner dans l'exercice
- Nous constatons qu'il y a deux cas triviaux $U_0 = 0$ et $U_1 = 1$

Solution récursive de Ex 2

Solution itérative

```
Int fibrecursive (int n)
{
   if (n <= 0)
        return  0;
   else
        if (n == 1)
             return 1;
         else return  fibrecursive (n-1) + fibrecursive (n-2);
}
```

Nous allons illustrer le fonctionnement de la fonction récursive sur une exemple pour n = 5

Solution récursive de Ex 2

Reprenons la formule utilisée dans la solution récursive

$U_0=0$;
$U_1=1$;
$U_n=U_{n-1}+U_{n-2}$ pour n>=2

Nous rappelons que lors de l'exécution, nous aurons besoin d'une pile pour garder trace des fonctions appelantes

$U_5 = U_4 + U_3 \longrightarrow$ on devrais calculer U_4 et U_3 pour pouvoir calculer $U_5 \longrightarrow$ ce qui provoque l' arrêt de U_5 et l'appel de $U_4 \longrightarrow$ On empile U_5

empiler

U_5

Pile vide

Solution récursive de Ex 2

$U_4 = U_3 + U_2$ on devrais calculer U_3 et U_2 pour pouvoir calculer U_4 $\longrightarrow$

ce qui provoque l' arrêt de U_4 et l'appel de U_3 $\longrightarrow$ On empile U_4

empiler

$\boxed{U_4}$

$\boxed{U_5}$

$U_3 = U_2 + U_1$ on devrais calculer U_2 et U_1 pour pouvoir calculer U_3 $\longrightarrow$

ce qui provoque l' arrêt de U_3 et l'appel de U_2 $\longrightarrow$ On empile U_3

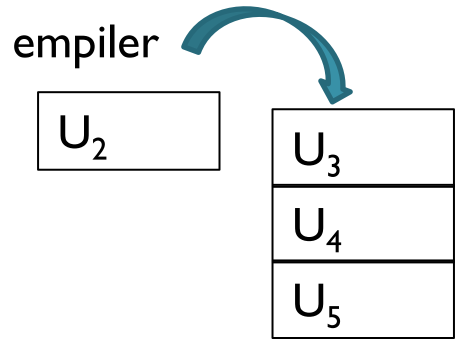
empiler

$\boxed{U_3}$

U_4
U_5

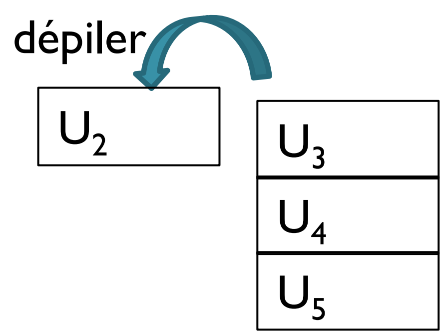
Solution récursive de Ex 2

$U_2 = U_1 + U_0$ on devrais calculer U_1 et U_0 pour pouvoir calculer U_2 $\longrightarrow$ ce qui provoque l' arrêt de U_2 et l'appel de U_1 $\longrightarrow$ on empile U_2



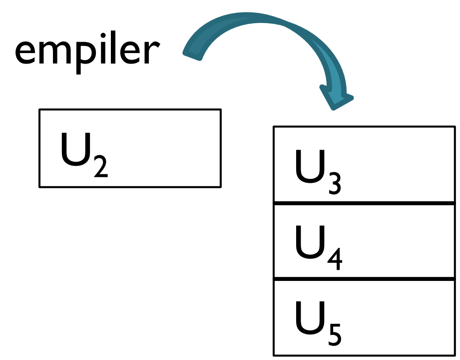
$U_1 = 1$ lors de l'appel de la fonction pour n = 1 $\longrightarrow$ où n = 1 un cas trivial $\longrightarrow$ donc la fonction se termine et retourne 1

- On dépile du sommet de la pile l' élément U_2



Solution récursive de Ex 2

- Revenons à la formule de $U_2 = U_1 + U_0$
- Après que U_1 ai terminé son exécution et de retour à U_2, on observe qu'il y a un appel à U_0 $\longrightarrow$ ce qui provoque l' arrêt de U_2 et l' exécution de U_0
- On empile U_2 dans la pile



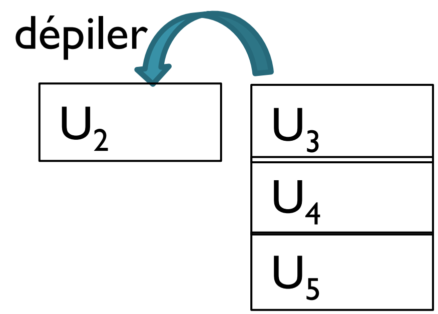
Rappel

La fonction U_n est une fonction récursive non terminale

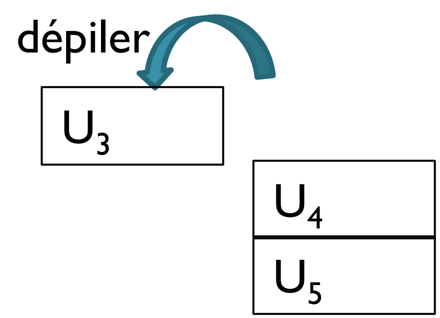
Solution récursive de Ex 2

$U_0 = 0$ lors de l'appel de la fonction pour $n = 0$ $\longrightarrow$ où $n = 0$ un cas trivial $\longrightarrow$ donc la fonction se termine et retourne 0

- On dépile du sommet de la pile l' élément U_2



- Revenons à la formule de $U_2 = U_1 + U_0$
- Nous avons maintenant la valeur de U_1 et U_0 $\longrightarrow$ donc U_2 peut se terminer $\longrightarrow$ elle retourne la valeur 1
- On dépile du sommet de la pile l' élément U_3

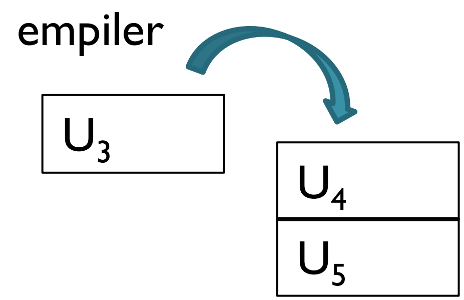


Solution récursive de Ex 2

- Revenons à la formule de $U_3 = U_2 + U_1$

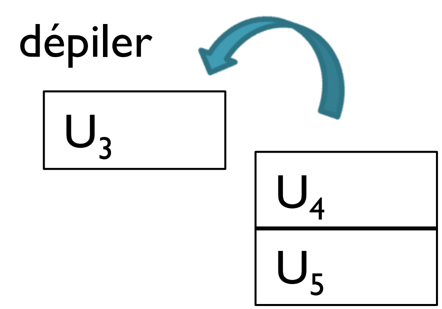
-Après que U_2 ai terminé son exécution et de retour à U_3, on observe qu'il y a un appel à U_1 → ce qui provoque l' arrêt de U_3 et l' exécution de U_1

- On empile U_3 dans la pile



$U_1 = 1$ lors de l'appel de la fonction pour n = 1 → où n = 1 un cas trivial → donc la fonction se termine et retourne 1

- On dépile du sommet de la pile l' élément U_3

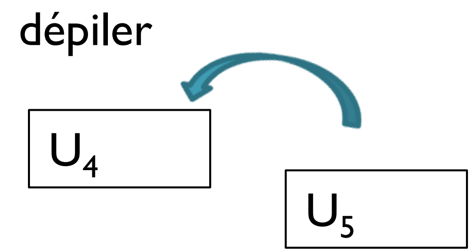


Solution récursive de Ex 2

- Revenons à la formule de $U_3 = U_2 + U_1$

- Nous avons maintenant la valeur de U_2 et U_1 $\longrightarrow$ donc U_3 peut se terminer $\longrightarrow$ elle retourne la valeur 2

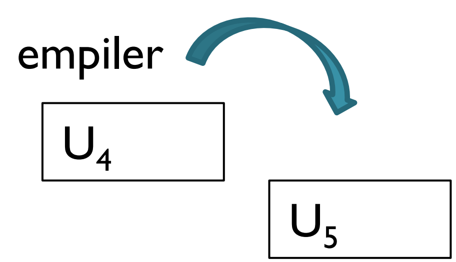
- On dépile du sommet de la pile l' élément U_4



- Revenons à la formule de $U_4 = U_3 + U_2$

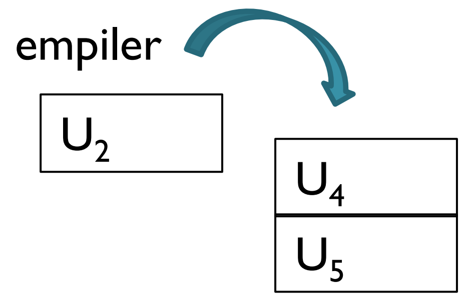
-Après que U_3 ai terminé son exécution et de retour à U_4, on observe qu'il y a un appel à U_2 $\longrightarrow$ ce qui provoque l' arrêt de U_4 et l' exécution de U_2

- On empile U_4 dans la pile



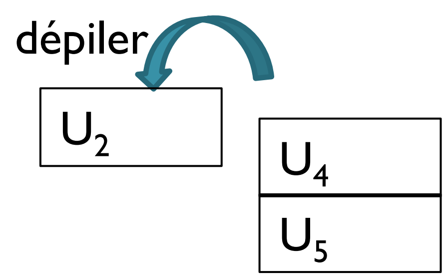
Solution récursive de Ex 2

$U_2 = U_1 + U_0$ on devrais calculer U_1 et U_0 pour pouvoir calculer U_2 $\longrightarrow$ ce qui provoque l' arrêt de U_2 et l'appel de U_1 $\longrightarrow$ on empile U_2



$U_1 = 1$ lors de l'appel de la fonction pour n = 1 $\longrightarrow$ où n = 1 un cas trivial $\longrightarrow$ donc la fonction se termine et retourne 1

- On dépile du sommet de la pile l' élément U_2

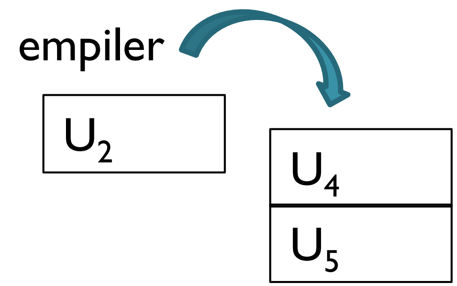


Solution récursive de Ex 2

- Revenons à la formule de $U_2 = U_1 + U_0$

-Après que U_1 ai terminé son exécution et de retour à U_2, on observe qu'il y a un appel à U_0 $\longrightarrow$ ce qui provoque l' arrêt de U_2 et l' exécution de U_0

- On empile U_2 dans la pile



$U_0 = 0$ lors de l'appel de la fonction pour $n = 0$ $\longrightarrow$ où $n = 0$ un cas trivial $\longrightarrow$ donc la fonction se termine et retourne 0

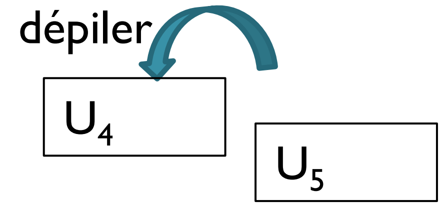
- On dépile du sommet de la pile l' élément U_2

-Revenons à la formule de $U_2 = U_1 + U_0$

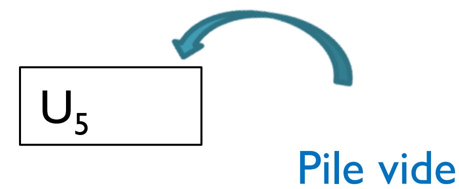
- Nous avons maintenant la valeur de U_1 et U_0 $\longrightarrow$ donc U_2 peut se terminer $\longrightarrow$ elle retourne la valeur 1

Solution récursive de Ex 2

- On dépile du sommet de la pile l’ élément U_4



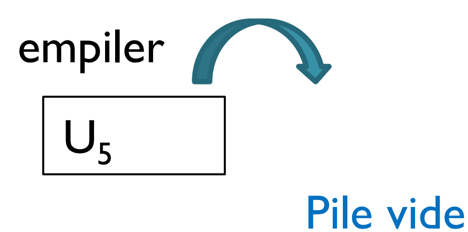
- Revenons à la formule de $U_4 = U_3 + U_2$
- Nous avons maintenant la valeur de U_3 et U_2 $\longrightarrow$ donc U_4 peut se terminer $\longrightarrow$ elle retourne la valeur 3
- On dépile du sommet de la pile l’ élément U_5



- Revenons à la formule de $U_5 = U_4 + U_3$

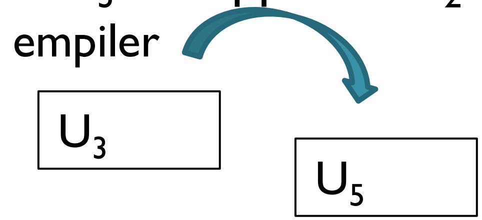
-Après que U_4 ai terminé son exécution et de retour à U_5, on observe qu’il y a un appel à U_3 $\longrightarrow$ ce qui provoque l’ arrêt de U_5 et l’ exécution de U_3

- On empile U_5 dans la pile

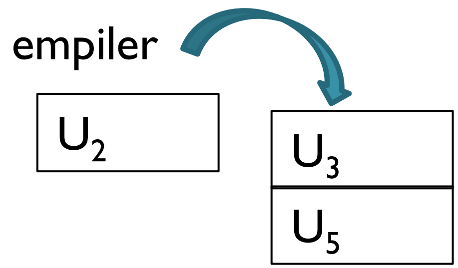


Solution récursive de Ex 2

$U_3 = U_2 + U_1$ on devrais calculer U_2 et U_1 pour pouvoir calculer U_3 → ce qui provoque l' arrêt de U_3 et l'appel de U_2 → On empile U_3

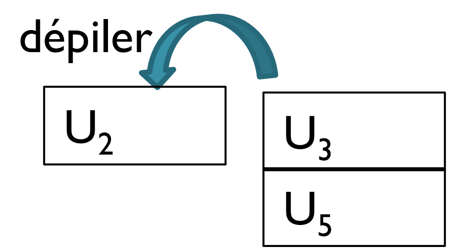


$U_2 = U_1 + U_0$ on devrais calculer U_1 et U_0 pour pouvoir calculer U_2 → ce qui provoque l' arrêt de U_2 et l'appel de U_1 → on empile U_2



$U_1 = 1$ lors de l'appel de la fonction pour n = 1 → où n = 1 un cas trivial → donc la fonction se termine et retourne 1

- On dépile du sommet de la pile l' élément U_2

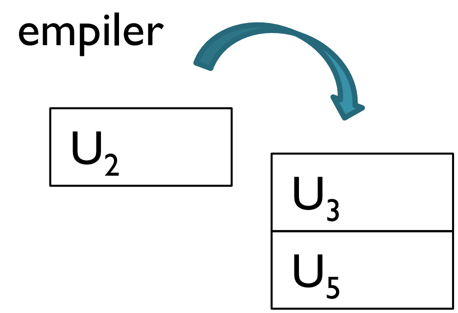


Solution récursive de Ex 2

- Revenons à la formule de $U_2 = U_1 + U_0$

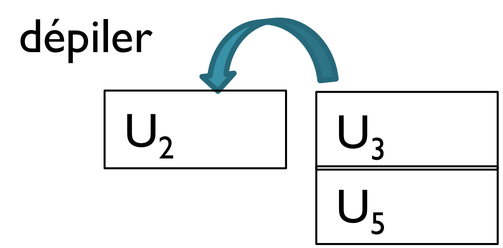
-Après que U_1 ai terminé son exécution et de retour à U_2, on observe qu'il y a un appel à U_0 → ce qui provoque l' arrêt de U_2 et l' exécution de U_0

- On empile U_2 dans la pile



$U_0 = 0$ lors de l'appel de la fonction pour n = 0 → où n = 0 un cas trivial → donc la fonction se termine et retourne 0

- On dépile du sommet de la pile l' élément U_2

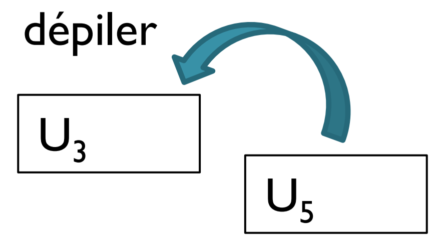


Solution récursive de Ex 2

- Revenons à la formule de $U_2 = U_1 + U_0$

- Nous avons maintenant la valeur de U_1 et U_0 $\longrightarrow$ donc U_2 peut se terminer $\longrightarrow$ elle retourne la valeur 1

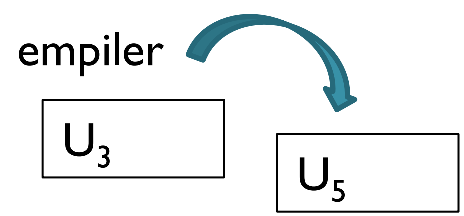
- On dépile du sommet de la pile l' élément U_3



- Revenons à la formule de $U_3 = U_2 + U_1$

-Après que U_2 ai terminé son exécution et de retour à U_3, on observe qu'il y a un appel à U_1 $\longrightarrow$ ce qui provoque l' arrêt de U_3 et l' exécution de U_1

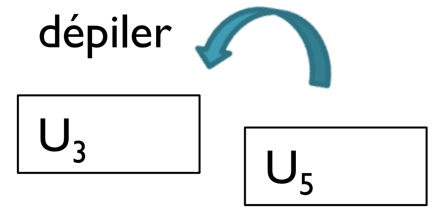
- On empile U_3 dans la pile



$U_1 = 1$ lors de l'appel de la fonction pour $n = 1$ $\longrightarrow$ où $n = 1$ un cas trivial $\longrightarrow$ donc la fonction se termine et retourne 1

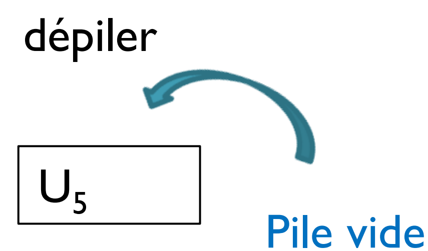
Solution récursive de Ex 2

- On dépile du sommet de la pile l' élément U_3



- Revenons à la formule de $U_3 = U_2 + U_1$

- Nous avons maintenant la valeur de U_2 et U_1 $\longrightarrow$ donc U_3 peut se terminer $\longrightarrow$ elle retourne la valeur 2



- On dépile du sommet de la pile l' élément U_5

- Revenons à la formule de $U_5 = U_4 + U_3$

- Nous avons maintenant la valeur de U_4 et U_3 $\longrightarrow$ donc U_5 peut se terminer $\longrightarrow$ elle retourne la valeur 5

Observations Ex 1 et Ex 2

- Les fonctions ont besoin d'un nombre entier n pour leurs calculs → nous l'avons mit autant que paramètre .
- La valeur de ce n sera fournit de l' extérieur de la fonction
- n est un paramètre passé par valeur → cela signifie que la fonction utilise sa valeur mais ne peux la changer.
- les variables déclarées à l' intérieur des fonctions → sont des variables locales à la fonction → ils sont crées lors de l'appel de la fonction et disparaissent à la fin de l'appel.