

1 Sans Ordinateurs !

Pour chacun des programmes proposés, vous vous poserez les questions suivantes :

- est-ce que le programme compile avec la commande "gcc" sans options ? Si oui y a-t-il des warnings ? Si oui lesquels ?
- lorsque l'on compile avec la commande "gcc -Wall -ansi -pedantic", y a-t-il des "warnings" supplémentaires ? si oui lesquels.
- identifiez dans ces programmes les déclarations et les définitions de variables et de fonctions.
- que font ces programmes ? Dessinez si nécessaire la progression de l'état de la pile.

1. Code 1

```
1 #include <stdio.h>
2 void f(int);
3 int main()
4 {
5     f(5);
6     return 0;
7 }
8 void f(int n)
9 {
10     if(n>0){
11         printf("%d\n",n);
12         f(n-1);
13     }
14 }
```

2. Code 2

```
1 #include <stdio.h>
2 void f(int);
3 int main()
4 {
5     f(5);
6     return 0;
7 }
8 void f(int n)
9 {
10     if(n>0){
11         f(n-1);
12         printf("%d\n",n);
13     }
14 }
```

3. Code 3

```
1 #include <stdio.h>
2 float f(int, int);
3 int main(){
4     float a;
5     a=3.0;
6     a=f(a, 3);
7     printf("%f\n", a);
```

```
8     return 0;
9 }
10 float f(int a, int b){
11     if(b<0)
12         return 1/f(a,-b);
13     if(b>0)
14         return a*f(a,b-1);
15     return 1;
16 }
```

2 1 étudiant par poste (si possible)

Commencez par recopier et tester les codes précédents pour vérifier vos intuitions. Pour la suite, vous écrirez une fonction `main()` unique que vous ferrez évoluer pour tester les différentes fonctions demandées.

1. Écrire une fonction récursive prenant en paramètre deux entiers `a` et `b` et affichant tout les entiers entre `a` et `b`. Par exemple si on l'appelle avec comme valeurs `a=3` et `b=6`, la fonction affiche 3 4 5 6.
2. Si ce n'est pas déjà le cas, modifiez votre fonction pour qu'elle affiche 6 5 4 3 si on l'appelle avec `a=6` et `b=3` ($a > b$).
3. Écrivez une fonction similaire à la précédente, toujours récursive, sauf qu'au lieu d'afficher les entiers, la fonction doit calculer leur somme et la renvoyer. Par exemple, si `a=3` et `b=5`, la fonction calcule $3+4+5$ et renvoie donc 12. Attention, pour tester votre fonction, vous devrez placer les affichages nécessaires dans le `main()`.
4. Tout ce qui peut s'écrire en programmation récursive peut s'écrire en programmation itérative, et vice versa. Il y a équivalence stricte entre ce qui est programmable en récursif, et ce qui est programmable en itératif, même si les performances des programmes ne seront pas, elles, équivalentes. Après avoir écouté les explications du chargé de td sur la boucle "for", refaites les fonctions précédentes sans utiliser la récursivité.