

Solution

Exercise 1

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

```
int produit(int a, int b)
{
    if (b == 0) return 0;
```

Cas trivial

```
    if (b % 2 == 0) return produit(a + a, b / 2);
    if (b % 2 == 1) return produit(a, b - 1) + a;
}
```

Cas récursive réappelle de la même fonction jusqu'à atteindre le cas trivial et retour de valeur final

```
void main()
{
    int A, B;
    printf("donner 2 entier\n");
    scanf("%d%d", &A, &B);
    printf("ton produit est %d", produit(A, B));
}
```

Appel de la fonction en respectant les type de paramètre (A et B ici sont des entier) et en respectant l'affichage du retour qui est entier aussi (%d)

Exemple d'exécution :

```
donner 2 entier
4
3
ton produit est 12
```

Exercise 2

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

```
int fibonaci(int n)
{
    if (n == 1 || n == 2) return 1;

    return fibonaci(n - 1) + fibonaci(n - 2);
}
```

Cas trivial

Cas récursive

```
void main()
{
    int n, U_n;
    printf("Saisir un entier : ");
    scanf("%d", &n);
    U_n = fibonaci(n);

    printf("L'entier U_%d = %d", n, U_n);
}
```

Affectation de U_n qui est de même nature du fonction int fibonaci

Exemple d'exécution :

```
Saisir un entier : 43
L'entier U_43 = 433494437
```

Exercice 3

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

```
void affichage()  
{  
    printf("\nSaisir 1 pour effectuer une addition\n Saisir 2  
pour effectuer une soustraction\n Saisir 3 pour effectuer une  
multiplication\n Saisir 4 pour effectuer une division le premier  
sur le deusieme\n Saisir 0 pour quitter le programme\n");  
}
```

Une fonction void est une fonction qui retourne rien. C'est un exemple du 'void' qui ne prend aucun paramètre et seulement affiche la liste d'options.

```
float add(float a, float b)  
{  
    return a + b;  
}
```

Fonction float qui return un nombre float additionnant 2 paramètres de type float aussi (a et b)

```
float sous(float a, float b)  
{  
    if (a > b) return a - b;  
    return b - a;  
}
```

Fonction float de soustraction absolue de 2 nombres. // on peut même mettre a-b non précisé dans l'exercice

```
float div(float a, float b)  
{  
    return a / b;  
}
```

Fonction float retournant une valeur de division float // à faire attention que b doit diffèrent que 0

```
float mult(float a, float b)  
{  
    return a*b;  
}
```

Fonction float retournant une valeur de multiplication float

```
float calcul(float a, float b, int n)  
{  
    switch (n)  
    {  
        case 1: return add(a, b);  
        case 2: return sous(a, b);  
        case 3: return mult(a, b);  
        case 4: return div(a, b);  
    }  
}
```

Une fonction qui prend en paramètre les deux nombres et un entier (choix ici) pour appliquer l'opération correspondante et retourner le bon résultat,

```
void lire(float a, float b, int n)  
{  
    switch (n)  
    {  
        case 0:  
            printf("Merci d'avoir utilise notre calculatrice, a bientôt !\n");  
            break;
```

Une fonction void qui analyse le choix de l'utilisateur et affiche selon l'option choisie.

```
        case 1: case 2: case 3: case 4:  
            printf("votre resultat est %f\n veuillez vous un autre tache saisir  
un option autre fois :\n", calcul(a, b, n));
```

Appelle afin d'affichage de la réponse correspondant selon n

```

scanf("%d", &n);
lire(a, b, n);
break;

default:
    printf("\nla valeur saisie ne correspond pas a une
tache reconnue par le programme !\n");
    affichage();
    scanf("%d", &n);
    lire(a, b, n);
    break;
}

void main()
{
    float a, b;
    int n;

    printf("donner deux nombres:\n");
    scanf("%f%f", &a, &b);

    affichage();
    scanf("%d", &n);

    lire(a, b, n);
}

```

Entrer dans la même fonction un cas de récursivité unique choisie après affichage du résultat

Entre dans la même fonction cas de récursivité lors de saisir d'un option incorrecte n différent des 'case' auparavant.

Prix de 3 paramètres

Appel du void comme instruction

Exemple d'exécution :

donner deux nombres:

3
7

Saisir 1 pour effectuer une addition

Saisir 2 pour effectuer une soustraction

Saisir 3 pour effectuer une multiplication

Saisir 4 pour effectuer une division le premier sur le deusieme

Saisir 0 pour quitter le programme

8

la valeur saisie ne correspond pas a une tache reconnue par le programme !

Saisir 1 pour effectuer une addition

Saisir 2 pour effectuer une soustraction

Saisir 3 pour effectuer une multiplication

Saisir 4 pour effectuer une division le premier sur le deusieme

Saisir 0 pour quitter le programme

2

votre resultat est 4.000000

veuiler vous un autre tache saisir un option autre fois :

0

Merci d'avoir utilise notre calculatrice, a bientot !

Exercice 4

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

```
void calcul_dose(int Ts[], float Tp[], float Td[], int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        if (Ts[i] == 0)
        {
            Td[i] = 0.5 / 4.5 * Tp[i];
        }
        else
        {
            Td[i] = 0.3 / 3.6 * Tp[i];
        }
    }
}
```

Calcul du tableau dose

Note sur les affectations dans les fonctions :

Jusqu'à maintenant dans notre cours tout paramètre (soit entier float...) reste in affecté après sortir de la fonction ;

Tandis qu'un tableau peut être affecter c.à.d. si on change les valeurs dans un tableau ce changement est sur tout le programme contraire aux autres variables des paramètres et de variable déclaré dans la fonction sont des variables locaux (variable local).

```
void calcul_pillule(float Td[], int Tpi[], int n)
{
    int Td_entier[N], i;
```

Calcul du tableau « pillule » à partir du dose

```
    for (i = 0; i < n; i++)
    {
        Td_entier[i] = Td[i];
    }
```

Réduction de tout variable float de la dose dans un autre tableau int c.à.d si on a 5.23 devient 5 c.à.d. tout après virgule sera ignoré

```
    for (i = 0; i < n; i++)
    {
        if (Td_entier[i] % 2 == 0) Tpi[i] = Td[i];
        else Tpi[i] = Td[i] - 1;
    }
}
```

Pour tout pair dans Tdose affecter son entier dans Tpillule
Pour tout impair dans Tdose affecter son entier moins un dans Tpillule

```
void affichage(int sexe[], int age[], float poids[], float dose[], int pillule[], int n)
{
```

```
    int i;
    printf("Sexe\tAge\tPoids\tDose\t\tPillule\n====\t===\t====\t====\t\t\t=====\n");
    for (i = 0; i < n; i++)
    {
        if (sexe[i] == 0) printf("H\t%d\t%.2f\t%.2f mg\t%d mg\n", age[i],
poids[i], dose[i], pillule[i]);
        else printf("F\t%d\t%.2f\t%.2f mg\t%d mg\n", age[i], poids[i], dose[i],
pillule[i]);
    }
}
```

Affichage

```

void main()
{
    int Tsexe[N], Tage[N], Tpillule[N], n, i;
    float Tpoids[N], Tdose[N];

    printf("Veuillez saisir le nombre de vos patients:\n");
    scanf("%d", &n);

    printf("Veuillez saisir le sexe (0 homme et 1 femme) \npuis l'age entier \npuis le
poids \nde chaque patient de vos %d patients:", n);

    for (i = 0; i < n; i++)
    {
        printf("\nle patient[%d] a comme(sexe,age,poids) :\n", i + 1);
        scanf("%d%d%f", &Tsexe[i], &Tage[i], &Tpoids[i]);
    }

    calcul_dose(Tsexe, Tpoids, Tdose, n);

    calcul_pillul(Tdose, Tpillule, n);

    affichage(Tsexe, Tage, Tpoids, Tdose, Tpillule, n);
}

```

Remplir le data
base des patients

Appel des fonctions...

Exemple d'exécution :

```

Veuillez saisir le nombre de vos patients:
4
Veuillez saisir le sexe (0 homme et 1 femme)
puis l'age entier
puis le poids
de chaque patient de vos 4 patients:
le patient[1] a comme(sexe,age,poids) :
1
26
50.12

le patient[2] a comme(sexe,age,poids) :
0
31
106.77

le patient[3] a comme(sexe,age,poids) :
1
23
43

le patient[4] a comme(sexe,age,poids) :
2
14
54

```

Sexe	Age	Poids	Dose	Pillule
F	26	50.12	4.18 mg	4 mg
H	31	106.77	11.86 mg	10 mg
F	23	43.00	3.58 mg	2 mg
F	14	54.00	4.50 mg	4