

TME Solo LU2IN002 2021

Le programme suivant contient des erreurs qui empêchent sa compilation ou son exécution correcte. Corrigez ces erreurs. **Attention**

: votre programme doit non seulement compiler (sans erreur ni avertissement), mais en plus s'exécuter correctement par rapport à ce qui est indiqué en commentaire.

```
/* Ce programme calcule la valeur de la constante d'Euler e
   avec une erreur relative inferieure a erreur */

#include <stdio.h>

void euler(float err) {
    int k = 0;
    float e = 0.0;
    float old_e;
    terme = 1.0;

    do {
        old_e = e;
        e = e + terme;
        k++;
        terme = terme * 1.0/k;
    } while ( (e - old_e)/e > err)

    return e;
}

int main() {
    float erreur;

    /* L'instruction suivante est correcte, elle permet de saisir
       la valeur de erreur */
    scanf("%f",&erreur);

    printf("e = %f\n",euler(erreur));
    return 0;
}
```

Question 2 :

Nous souhaitons écrire un programme calculant le coût du tirage et de l'envoi de photos.

Le coût du tirage par photo est de:

- 0,07 euro de 1 à 149 photos
- 0,06 euro de 150 à 749 photos
- 0,05 euro à partir de 750 photos

Les frais de port sont de:

- 4 euros jusqu'à 200 photos
- 8 euros de 201 à 500 photos
- 10 euros à partir de 501 photos

Les différentes valeurs du coût de tirage et des frais de port sont définies par des directives `#define`.

Complétez les fonctions :

- `prixPhotos`, qui prend en paramètre la quantité de photos à tirer et renvoie le prix, port compris, du tirage. Nous faisons l'hypothèse que le paramètre est un entier strictement positif, ce n'est pas à vérifier.
- `main`, pour que le prix soit affiché (vous devez remplacer les ... dans le dernier appel à `printf`)

```
#include <stdio.h>
#define PR1X149 0.07
#define PR1X749 0.06
#define PR1X 0.05
#define PORT200 4
#define PORT500 8
#define PORT 10

... prixPhotos(...) {

}

int main() {
    int quantite;

    /* L'instruction suivante permet de saisir la valeur de la variable quantite.
```

```

    Vous ne devez pas la modifier */
    scanf("%d",&quantite);
    printf("Quantite : %d\n",quantite);

    printf("Prix (port compris) : %.2f\n",...);
    return 0;
}

```

Question 3 :

Nous souhaitons que le programme suivant affiche la somme des entiers strictement inférieurs à une valeur donnée parmi `n` valeurs. La fonction `valeur`, qui vous est donnée, renvoie une nouvelle valeur entière.

Complétez les fonctions :

- `somme` qui prend en paramètre deux entiers correspondant au nombre de valeurs à considérer et la borne. La fonction récupère le nombre de valeurs demandées (en appelant la fonction `valeur`) et renvoie la somme des valeurs strictement inférieures à la borne (0 si toutes les valeurs sont supérieures ou égales à la borne).
- `main` qui affiche le résultat attendu du programme (vous devez remplacer les ... du dernier appel à `printf`).

```

#include <stdio.h>

/* Vous ne devez pas modifier cette fonction */
int valeur() {
    int n;
    scanf("%d",&n);
    return n;
}

... somme(...) {

}

int main() {
    int borne, n;

    /* Les deux instructions suivantes permettent la saisie de la valeur des
    variables borne et n.
    Vous ne devez pas les modifier. */
    scanf("%d",&borne);

```

```

scanf("%d",&n);
printf("Borne : %d, Nb valeurs : %d\n",borne,n);

printf("somme = %d\n",...);
return 0;
}

```

Question 4 :

La fonction `valeur` qui vous est donnée renvoie une valeur entière.

Complétez les fonctions

- `ma_fonction` qui prend au moins un entier en paramètre, `n`, et qui permet de récupérer la somme et la somme des carrés des `n` entiers obtenus chacun par un appel à `valeur`. Nous faisons l'hypothèse que `n` est strictement positif, ce n'est pas à vérifier.
- `main` en complétant les appels aux fonction `ma_fonction` et `afficherResultat`. Vous ne devez ajouter ou modifier aucune autre instruction.

Vous ne devez pas modifier la fonction `afficherResultat`

```

#include <stdio.h>

/* Vous ne devez pas modifier cette fonction */
int valeur(void) {
    int n;

    scanf("%d",&n);
    return n;
}

/* Vous ne devez pas modifier le type de retour de la fonction */
void ma_fonction(...) {

}

void afficherResultat(int somme, int somme_carres) {
    printf("Somme=%d, Somme carres=%d\n",somme,somme_carres);
}

/* Vous devez completer la fonction main suivante sans modifier les declarations
   et instructions
   qui vous sont donnees */
int main() {

```

```

int nb;
int somme, somme_carres;

/* Instruction permettant la saisie de la valeur de nb */
scanf("%d",&nb);

/* A COMPLETER, remplacez les pointilles */
ma_fonction(...);

/* A COMPLETER, remplacez les pointilles */
afficherResultat(...);

return 0;
}

```

Corrigé (Etudiant 20/20)

Question 1 :

```

/* Ce programme calcule la valeur de la constante d'Euler e
   avec une erreur relative inferieure a erreur */

#include <stdio.h>

float euler(float err) {
    int k = 0;
    float e = 0.0;
    float old_e;
    float terme = 1.0;

    do {
        old_e = e;
        e = e + terme;
        k++;
        terme = terme * 1.0/k;
    } while ( (e - old_e)/e > err);
    return e;
}

int main() {
    float erreur;

```

```

/* L'instruction suivante est correcte, elle permet de saisir
la valeur de erreur */
scanf("%f",&erreur);

printf("e = %f\n",euler(erreur));
return 0;
}

```

Question 2 :

```

#include <stdio.h>
#define PR1X149 0.07
#define PR1X749 0.06
#define PR1X 0.05
#define PORT200 4
#define PORT500 8
#define PORT 10

float prixPhotos(int nbPhoto) {
    float prix = 0.0;
    /* Calcul du coût du tirage */
    if(nbPhoto > 149) {
        if(nbPhoto > 749) {
            prix = nbPhoto*PR1X;
        } else {
            prix = nbPhoto*PR1X749;
        }
    } else {
        prix = nbPhoto*PR1X149;
    }
    /* Calcul des frais de port */
    if(nbPhoto > 200) {
        if(nbPhoto > 500) {
            prix += PORT;
        } else {
            prix += PORT500;
        }
    } else {
        prix += PORT200;
    }
    return prix;
}

int main() {
    int quantite;

    /* L'instruction suivante permet de saisir la valeur de la variable quantite.
    Vous ne devez pas la modifier */
    scanf("%d",&quantite);
}

```

```

printf("Quantite : %d\n",quantite);

printf("Prix (port compris) : %.2f\n",prixPhotos(quantite));
return 0;
}

```

Question 3 :

```

#include <stdio.h>

/* Vous ne devez pas modifier cette fonction */
int valeur() {
    int n;
    scanf("%d",&n);
    return n;
}

int somme(int *nbVal,int *b) {
    int s = 0;
    for(int i = 0; i < *nbVal; i++) {
        int temp = valeur();
        if(temp < *b) {
            s += temp;
        }
    }
    return s;
}

int main() {
    int borne, n;

    /* Les deux instructions suivantes permettent la saisie de la valeur des
    variables borne et n.
    Vous ne devez pas les modifier. */
    scanf("%d",&borne);
    scanf("%d",&n);
    printf("Borne : %d, Nb valeurs : %d\n",borne,n);

    printf("somme = %d\n",somme(&n,&borne));
    return 0;
}

```

Question 4 :

```

#include <stdio.h>

```

```

/* Vous ne devez pas modifier cette fonction */
int valeur(void) {
    int n;

    scanf("%d",&n);
    return n;
}

/* Vous ne devez pas modifier le type de retour de la fonction */
void ma_fonction(int n, int *s, int *sC) {
    for(int i = 0; i < n; i++) {
        int temp = valeur();
        *s += temp;
        *sC += temp*temp;
    }
}

void afficherResultat(int somme, int somme_carres) {
    printf("Somme=%d, Somme carres=%d\n",somme,somme_carres);
}

/* Vous devez completer la fonction main suivante sans modifier les declarations
   et instructions
   qui vous sont donnees */
int main() {
    int nb;
    int somme = 0, somme_carres = 0;

    /* Instruction permettant la saisie de la valeur de nb */
    scanf("%d",&nb);

    /* A COMPLETER, remplacez les pointilles */
    ma_fonction(nb,&somme,&somme_carres);

    /* A COMPLETER, remplacez les pointilles */
    afficherResultat(somme,somme_carres);

    return 0;
}

```

Corrigé (Auteur du TME)

Question 1 :


```

/* Ce programme calcule la valeur de la constante d'Euler e
   avec une erreur relative inferieure a erreur */

#include <stdio.h>

float euler(float err) {
    int k = 0;
    float e = 0.0, old_e, terme = 1.0;

    do {
        old_e = e;
        e = e + terme;
        k++;
        terme = terme * 1.0/k;
    } while ( (e - old_e)/e > err);

    return e;
}

int main() {
    float erreur;

    /* L'instruction suivante est correcte, elle permet de saisir
       la valeur de erreur */
    scanf("%f",&erreur);

    printf("e = %f\n", euler(erreur));
    return 0 ;
}

```

Question 2 :

```

#include <stdio.h>
#define PR1X149 0.07
#define PR1X749 0.06
#define PR1X 0.05
#define PORT200 4
#define PORT500 8
#define PORT 10

float prixPhotos(int quantite) {
    float prix;

    if (quantite <= 149) {
        prix = quantite * PR1X149;
    } else {
        if (quantite <= 749) {
            prix = quantite * PR1X749;

```

```

    } else {
        prix = quantite * PRIX;
    }
}

if (quantite <= 200) {
    prix = prix + PORT200;
} else {
    if (quantite <= 500) {
        prix = prix + PORT500;
    } else {
        prix = prix + PORT;
    }
}

return prix;
}

int main() {
    int quantite;

    /* L'instruction suivante permet de saisir la valeur de la variable quantite.
    Vous ne devez pas la modifier */
    scanf("%d",&quantite);
    printf("Quantite : %d\n",quantite);

    printf("Prix (port compris) : %.2f\n",prixPhotos(quantite));
    return 0;
}

```

Question 3 :

```

#include <stdio.h>

/* Vous ne devez pas modifier cette fonction */
int valeur() {
    int n;
    scanf("%d",&n);
    return n;
}

int somme(int borne, int nb) {
    int res=0, val,i;

    for (i=0; i < nb; i++) {
        val=valeur();
        if (val < borne) {
            res = res + val;
        }
    }
}

```

```

    return res;

}

int main() {
    int borne, n;

    /* Les deux instructions suivantes permettent la saisie de la valeur des
       variables borne et n.
       Vous ne devez pas les modifier. */
    scanf("%d",&borne),
    scanf("%d",&n);
    printf("Borne : %d, Nb valeurs : %d\n",borne,n);

    printf("somme = %d\n",somme(borne,n));
    return 0;
}

```

Question 4 :

```

#include <stdio.h>

/* Vous ne devez pas modifier cette fonction */
int valeur(void) {
    int n;

    scanf("%d",&n);
    return n;
}

/* Vous ne devez pas modifier le type de retour de la fonction */
void ma_fonction(int nb, int *somme, int *somme_carres) {
    int i;
    int v;
    *somme=0;
    *somme_carres=0;

    for (i=0; i < nb; i++) {
        v=valeur();
        *somme = *somme + v;
        *somme_carres = *somme_carres + v*v;
    }
}

void afficherResultat(int somme, int somme_carres) {
    printf("Somme=%d, Somme carres=%d\n",somme,somme_carres);
}

```

```
/* Vous devez completer la fonction main suivante sans modifier les declarations
   et instructions
   qui vous sont donnees */
int main() {
    int nb;
    int somme, somme_carres;

    /* Instruction permettant la saisie de la valeur de nb */
    scanf("%d",&nb);

    /* A COMPLETER, remplacez les pointilles */
    ma_fonction(nb,&somme,&somme_carres);

    /* A COMPLETER, remplacez les pointilles */
    afficherResultat(somme,somme_carres);

    return 0;
}
```