

Examen 2^{ème} Semestre 2019-2020

Matière	Langage C avancé
Durée	1 h30
Niveau	PI1
Enseignant	Youssef ALJ

Documents autorisés

Consignes :

1. Les exercices sont indépendants et peuvent être traités dans l'ordre souhaité.
2. Mentionner le numéro de l'exercice ainsi que celui de la question traitée.

Exercice 1 (4 points):

Trouver les 8 erreurs dans le programme suivant :

```
#include <stdio.h>
int main() {
    int a;
    char b; c;
    printf("bonjour, \n");
    printf("veuillez saisir deux caractères ");
    scanf("%c", &b);
    scanf("%c", c);
    printf("le caractère saisi est %c\n", &b);
    printf("saisir une valeur entière: ");
    scanf("%d", &a);
    printf("vous avez saisi \n", a);
    printf("au revoir.\n");
    return 0;
}
```

Exercice 2 (9 points) :

On demande à une classe d'écrire un programme qui calcule la somme des nombres pairs de 0 jusqu'à un nombre n défini par l'utilisateur. Deux étudiants Katy et Ahmed proposent leur programme (voir programmes ci-dessous), mais ils n'arrivent pas à remplir certaines parties laissées en pointillés

On se propose dans les questions qui suivent d'aider les deux étudiants.

Partie A : aidons Ahmed (voir programme d'Ahmed ci-dessous):

Ahmed propose de répondre au problème posé en parcourant tous les entiers de 0 jusqu'au nombre n défini par l'utilisateur et tester à chaque fois si cet entier est pair.

1. Pouvez-vous aider Ahmed à remplir les lignes 11 et 12 de son programme?
2. Dans la ligne 13 de son programme, Ahmed souhaite calculer la somme demandée en mettant à jour la variable `somme` déclarée dans la ligne 5. Pouvez-vous l'aider à remplir la ligne 13 de son programme?

Partie B : aidons Katy (voir programme de Katy ci-dessous):

Katy, quant à elle, propose d'organiser son programme sous forme de fonctions. Elle déclare une fonction `somme_pairs` qui prend comme paramètre un entier `n` qui représente le nombre défini par l'utilisateur.

De plus, elle prétend qu'elle n'a pas besoin d'un test `if` à l'intérieur de la boucle `for` comme suggéré par Ahmed.

3. Pouvez-vous aider Katy en remplissant la définition de la fonction `somme_pairs` ci-dessous?
4. Comment appeler la fonction `somme_pairs` dans la ligne 12 pour que le programme de Katy fonctionne correctement?

Programme d'Ahmed	
1	<code>#include <stdio.h></code>
2	
3	<code>int main() {</code>
4	<code> int i, n;</code>
5	<code> int somme=0;</code>
6	<code> printf("veuillez saisir un entier\n");</code>
7	<code> scanf("%d", &n);</code>
8	<code> // on parcourt les nombres de 0 jusqu'a n</code>
9	<code> // si on tombe sur un nombre pair</code>
10	<code> // on le rajoute a la somme</code>
11	<code> for (.....){</code>
12	<code> if (.....){</code>
13	<code> </code>
14	<code> }</code>
15	<code> }</code>
16	<code> printf("somme des pairs jusqu'à %d est %d", n, somme);</code>
17	<code> return 0;</code>
18	<code>}</code>

Programme de Katy	
1	<code>#include <stdio.h></code>
2	
3	<code>// delcaration de la fonction</code>
4	<code>int somme_pairs(int n);</code>
5	
6	<code>int main() {</code>
7	<code> int n;</code>
8	<code> printf("veuillez saisir un entier\n");</code>
9	<code> scanf("%d", &n);</code>
10	<code> int somme;</code>
11	<code> // appel de la fonction</code>
12	<code> </code>
13	<code> printf("somme des pairs jusqu'à %d est %d", n, somme);</code>
14	<code> return 0;</code>
15	<code>}</code>
16	
17	<code>//definition de la fonction</code>
18	<code>int somme_pairs(int n){</code>
19	<code> int i, somme = 0;</code>
20	<code> for (.....){</code>
21	<code> </code>
22	<code> }</code>
23	<code> return ...;</code>
24	<code>}</code>

Partie C : analyse des deux programmes :

5. Concernant le choix de Katy d'utiliser une fonction, pensez-vous que ce choix est meilleur que celui d'Ahmed? Pourquoi?
6. Concernant le choix de Katy de ne pas utiliser un test `if` dans la boucle `for`, pensez-vous que ce choix est meilleur que celui d'Ahmed? Pourquoi?

Exercice 3 (7 points + 2 points bonus):

Ecrire un programme en C qui fait les opérations suivantes :

1. déclare trois variables `a`, `b` et `c` de type entier.
2. demande à l'utilisateur de saisir les valeurs de `a` et de `b`.
3. lit ce que l'utilisateur a saisi au clavier pour les valeurs `a` et `b`.
4. crée deux pointeurs `ptr_a` et `ptr_b` qui pointent sur `a` et `b` respectivement.

On rajoute après les opérations précédentes les lignes suivantes :

```
c = *ptr_a;  
*ptr_a = *ptr_b;  
*ptr_b = c;  
printf("les valeurs de a et b sont %d %d", *ptr_a, *ptr_b);
```

On suppose que le programme compile correctement, et à l'exécution l'utilisateur a saisi la valeur de 1 pour `a`, et la valeur de 2 pour `b`.

5. Quel résultat voit-on à l'écran à la fin de l'exécution de ce programme?
6. En une phrase, décrire ce que fait ce programme.

Question bonus (à traiter une fois toutes les questions de l'examen ont été traitées) :

Ré-organiser le programme précédent sous forme de fonctions tout en respectant le comportement voulu à la question 6.