

QCM

Langage C

QCM Langage C
Semestre 5
Examen du 09/01/2020

Nom et prénom :

SADURNI Thomas*Durée : 40 minutes.**Aucun document n'est autorisé. L'usage de la calculatrice est interdit.**Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.**Une question simple rapporte au maximum 1 point, une question multiple au maximum 3 points.**Des points négatifs pourront être affectés à de très mauvaises réponses.**Pour valider un choix, il faut cocher la case.***Seules les cases sont analysées,***il vous est donc possible d'écrire ailleurs sans incidence sur votre rendu.*

1 Structure d'un programme

On considère le programme suivant qui calcule le périmètre d'un carré connaissant son côté :

```
1  #include <stdlib.h>
2  int main(){
3      float cote = 2.0;    // longueur du côté
4      char unite = 'm';
5      printf("Le périmètre du carré de côté %1.0f%c est : ", cote, unite);
6      -- calcul du périmètre du carré
7      float perimetre = 4*cote;
8      printf("%1.2f%c\n", perimetre, unite);
9      return EXIT_OK;
10 }
```

Question 1 ♣ Quelle(s) instruction(s) ne sont pas correctes :

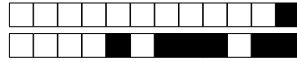
☒ L'instruction de la ligne 12.☐ L'instruction de la ligne 9.☒ L'instruction de la ligne 2.☒ L'instruction de la ligne 6.☐ Aucune de ces réponses n'est correcte.

Question 2 La compilation de ce programme affiche l'avertissement suivant :

```
perimetre.c: In function 'main':
perimetre.c:5:5: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
    printf("Le périmètre du carré de côté %1.0f%c est : ", cote, unite);
    ~~~~~
perimetre.c:5:5: warning: incompatible implicit declaration of built-in function 'printf'
```

Que faire pour supprimer cette erreur ?

☒ Inclure la bibliothèque `string` avec `#include <string.h>`☐ Corriger le descripteur de format `%1.0f` en `%1.0d` à la ligne 5.☐ Supprimer l'inclusion `#include <stdlib.h>`☒ Corriger le descripteur de format `%c` en `%s` à la ligne 5.☐ Inclure la bibliothèque `stdio` avec `#include <stdio.h>`



2 Constantes et types

Question 3 Comment définir la constante préprocesseur MAX qui vaut 10 ?

- | | |
|--|--|
| <input type="checkbox"/> #const MAX 10 | <input checked="" type="checkbox"/> #define MAX = 10 |
| <input checked="" type="checkbox"/> #define MAX 10 | <input type="checkbox"/> #const MAX = 10; |
| <input type="checkbox"/> #define MAX = 10; | <input type="checkbox"/> #define MAX 10; |

Question 4 Comment définir Entier comme un alias sur le type `int` ?

- | | |
|--|--|
| <input type="checkbox"/> typedef Entier int; | <input checked="" type="checkbox"/> #define Entier int |
| <input type="checkbox"/> typedef int Entier; | <input type="checkbox"/> type Entier is int; |

Question 5 Comment déclarer un nouveau type `t_tab` qui est un tableau de 20 réels ?

- | | |
|--|---|
| <input checked="" type="checkbox"/> define double[20] t_tab; | <input checked="" type="checkbox"/> typedef double t_tab[20]; |
| <input type="checkbox"/> define double t_tab[20]; | <input type="checkbox"/> typedef double t_tab 20; |

3 Variables et pointeurs

Question 6 Comment déclarer une variable `n` de type entier ?

- | | |
|----------------------------------|------------------------------------|
| <input type="checkbox"/> int n; | <input type="checkbox"/> n of int; |
| <input type="checkbox"/> int n | <input type="checkbox"/> n: int |
| <input type="checkbox"/> n: int; | |

Question 7 ♣ Comment déclarer et initialiser une variable `x` de type `double` à 20 ?

- | | |
|--|---|
| <input type="checkbox"/> double x := 20.0; | <input type="checkbox"/> double x = 20.0; |
| <input type="checkbox"/> double x = 20; | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |
| <input type="checkbox"/> double x += 20.0; | |

Question 8 L'instruction `reste -= diviseur;` permet de :

- ☐ soustraire la valeur de `diviseur` à la valeur de `reste`, et d'affecter le résultat à `reste`
- ☐ soustraire la valeur de `reste` à `diviseur` et d'affecter le résultat à `reste`

Question 9 Quelles valeurs donner à `XX1`, `XX2` et `XX3` pour que le message « Bravo ! » s'affiche lors de l'exécution des instructions suivantes :

```
assert(XX1 == '5' - '0');  
assert(XX2 == '0' + 7);  
assert(XX3 == 'A' + 3);  
printf("%s", "Bravo !");
```

- | | |
|--|--|
| <input type="checkbox"/> XX1='5', XX2='7' et XX3='D' | <input type="checkbox"/> XX1=5, XX2='7' et XX3='D' |
| <input type="checkbox"/> XX1=5, XX2=7 et XX3='C' | <input type="checkbox"/> XX1='5', XX2=7 et XX3='D' |

Question 10 ♣ Comment déclarer deux variables `a` et `b` de type pointeur sur caractère ?

- | | |
|--|---|
| <input type="checkbox"/> char* a, b; | <input type="checkbox"/> char *a, *b; |
| <input type="checkbox"/> char * a, * b; | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |
| <input type="checkbox"/> char* a; char *b; | |



Question 11 On suppose les instructions suivantes :

```
int var1 = 10;
int *p1 = &var1;
int *p2 = p1;
```

Quelle est la valeur de la variable p2 ?

☐ 10☐ l'adresse de p1.☐ l'adresse de var1.

Question 12 On suppose les instructions suivantes :

```
int var1 = 10;
int *p1 = &var1;
*p1 = 25;
printf("%d", var1);
```

Qu'affiche la dernière instruction ?

☐ 25☐ l'adresse de var1☐ 10

Question 13 ♣ Cet exercice s'intéresse au concept de masquage des variables. Soit le programme suivant (les inclusions de bibliothèque sont volontairement omises) :

```
1  int main() {
2      int alea = 20;
3      int diviseur = 2;
4      {
5          int alea = 3;
6          float diviseur = 2.0;
7          float res = alea / diviseur;
8          assert(res == 1.5);
9      }
10     int res = alea / diviseur;
11     assert(res == 10);
12
13     printf("%s", "Les tests passent\n");
14     return EXIT_SUCCESS;
15 }
```

Quelle(s) variable(s) sont masquée(s) dans cet exemple ?

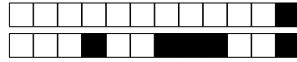
☐ int res à la ligne 10 est masquée.☐ int alea à la ligne 5 est masquée.☐ int alea à la ligne 2 est masquée.☐ float res à la ligne 7 est masquée.☐ int diviseur à la ligne 3 est masquée.☐ Aucune de ces réponses n'est correcte.☐ float diviseur à la ligne 6 est masquée.

4 Entrées/sorties

Question 14 ♣ La fonction `printf` permet d'écrire des données typées à l'écran. Dans la suite, cocher les utilisations correctes de cette fonction (i.e. qui ne génèrent pas d'avertissement à la compilation). On supposera que les variables suivantes sont définies et initialisées comme suit :

```
float prix = 2.5; int val = 12; char unite = 'm'; float peri = 22.3;
```

☐ `printf("%.2f%c", unite, peri);`☐ `printf("L'unité est %s", unite);`☐ `printf("Le prix est %f euros", prix);`☐ `printf("val = %d", &val);`☐ `printf('Le prix est %d euros', prix);`☐ Aucune de ces réponses n'est correcte.



Question 15 ♣ La fonction `scanf` permet de lire des données typées entrées au clavier. Dans la suite, cocher les utilisations correctes de cette fonction. On supposera que les variables suivantes sont définies comme suit :

```
float prix; int val; char unite = 'm'; float peri; char nom[10];
```

- | | |
|---|---|
| <input type="checkbox"/> <code>scanf("%1.2d", peri);</code> | <input type="checkbox"/> <code>scanf("%lf", &val);</code> |
| <input type="checkbox"/> <code>scanf("%d %c", &val, &unite);</code> | <input type="checkbox"/> <code>scanf("%s", &nom);</code> |
| <input type="checkbox"/> <code>scanf("%s", nom);</code> | <input type="checkbox"/> <code>scanf("Le prix est %f", &prix);</code> |
| <input type="checkbox"/> <code>scanf("%s", &unite);</code> | <input type="checkbox"/> <i>Aucune de ces réponses n'est correcte.</i> |

5 Structures de contrôle

Question 16 On suppose que `sequence1` et `sequence2` représentent plusieurs instructions où chaque instruction se termine par un point-virgule. On suppose aussi que `cond` est une expression booléenne. La conditionnelle :

Si `cond` Alors `sequence1` Sinon `sequence2`

peut alors s'écrire en C :

- ☐ `if (cond) then { sequence1 } else { sequence2 }`
- ☐ `if (cond) { sequence1 } else { sequence2 }`
- ☐ `if cond { sequence1 } else { sequence2 }`

Question 17 ♣ Quelle(s) formulation(s) de la boucle `pour` sont juste(s) ? On supposera que `sequence` représente plusieurs instructions.

- ☐ `for (int i = 0; i < 100; i++) sequence`
- ☐ `for (int j = 0; j <= 10 ; j += 2){ sequence }`
- ☐ `for (int i = 0; i < 100; i++){ sequence }`
- ☐ `for (int j = 1; j +=2 ; j < 10){ sequence }`
- ☐ `for (int k = 100; k >= 0; k--){ sequence }`
- ☐ `for (int i = 0, i < 10, i += 2){ sequence }`
- ☐ *Aucune de ces réponses n'est correcte.*

**Question 18 ♣**

Voici une fonction qui illustre la conditionnelle Selon :

```
int f(int n) {
    int r = 0;

    switch (n) {
        case 1:
            r += 1;
            break;
        case 2:
        case 3:
            r += 8;
            break;
        case 4:
            r += 99;
            break;
        case 5:
        case 7:
            r += 10;
        case 10:
        case 13:
            r += 100;
        default:
            r -= 1;
    }
    return r;
}
```

Elle retourne une valeur de `r` qui dépend du paramètre `n`. Quels tests sont corrects parmi les propositions suivantes :

- | | |
|---|--|
| <input type="checkbox"/> <code>assert(99 == f(10));</code> | <input type="checkbox"/> <code>assert(0 == f(0));</code> |
| <input type="checkbox"/> <code>assert(100 == f(13));</code> | <input type="checkbox"/> <code>assert(-1 == f(12));</code> |
| <input type="checkbox"/> <code>assert(8 == f(3));</code> | <input type="checkbox"/> <i>Aucune de ces réponses n'est correcte.</i> |

6 Enumération, Enregistrement et Tableau

Question 19 On souhaite définir un type énuméré qui représente les enseignes d'un jeu de cartes pique, coeur, carreau, trèfle. Quelle proposition retiendriez-vous pour cela ?

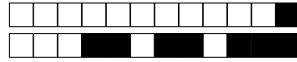
- ☐ `enum Enseigne is new {PIQ, COE, CAR, TRE};`
- ☐ `Enseigne is new enum (PIQ; COE; CAR; TRE);`
- ☐ `Enseigne is enum {PIQ, COE, CAR, TRE};`
- ☐ `enum Enseigne {PIQ, COE, CAR, TRE};`

Question 20 ♣ Quelles propositions sont vraies pour le type `enum Enseigne` défini à la question précédente :

- ☐ L'instruction `enum Enseigne ens = 0;` ne compile pas.
- ☐ `PIQ`, `COE`, `CAR`, `TRE` sont des constantes entières.
- ☐ On a la relation d'ordre suivante : `PIQ < COE < CAR < TRE`.
- ☐ *Aucune de ces réponses n'est correcte.*

Question 21 On souhaite définir un type enregistrement qui représente une carte à jouer caractérisée par une enseigne et une valeur entière. Quelle proposition retiendriez-vous pour cela ?

- ☐ `struct Carte is new {enum Enseigne ens; int val;};`
- ☐ `Carte is new struct (enum Enseigne ens; int val;);`
- ☐ `struct Carte { enum Enseigne ens; int val; };`
- ☐ `Carte is struct (enum Enseigne ens; int val;);`



Question 22 ♣ Le pointeur `ptr_carte` est initialisé de la façon suivante :

```
1 struct Carte carte1 = {PIQ, 9};  
2 struct Carte * ptr_carte = &carte1;
```

On souhaite afficher la valeur de `carte1` via le pointeur `ptr_carte`. Quelle(s) instruction(s) permet(tent) de le faire ?

- ☐ `printf("%d", ptr_carte -> val);`
- ☐ `printf("%d", ptr_carte.All.val);`
- ☐ `printf("%d", (*ptr_carte).val);`
- ☐ `printf("%d", ptr_carte.val);`
- ☐ *Aucune de ces réponses n'est correcte.*

Question 23 On souhaite définir un **type** tableau qui représente un jeu de 52 cartes à jouer. Quelle proposition retiendriez-vous pour cela ?

- ☐ `struct Carte Jeu[52];`
- ☐ `typedef struct Carte[52] Jeu;`
- ☐ `typedef struct Carte Jeu[52];`

Question 24 ♣ On souhaite initialiser la première carte d'un jeu avec l'as de pique. On suppose que la variable `jeu1` est déclarée comme suit `Jeu jeu1;`. Quelles propositions sont possibles ?

- ☐ `jeu1->ens = PIQ; jeu1->val = 1;`
- ☐ `jeu1[1].ens = PIQ; jeu1[1].val = 1;`
- ☐ `jeu1(0).ens = PIQ; jeu1(0).val = 1;`
- ☐ `jeu1[0].ens = PIQ; jeu1[0].val = 1;`
- ☐ `jeu1(1).ens = PIQ; jeu1(1).val = 1;`
- ☐ *Aucune de ces réponses n'est correcte.*

7 Sous-programmes

Question 25 Quelle est la signature qui correspond à une procédure `p` qui prend comme premier paramètre un entier `n` en entrée et comme deuxième paramètre un entier `d` en entrée/sortie.

- ☐ `void p(int n, int *d);`
- ☐ `void p(int* n, int *d);`
- ☐ `void p(int n, int d);`
- ☐ `int p(int n, int d);`
- ☐ `void p(int n, *d);`



Question 26 ♣ On considère le bout de code suivant :

```
void p1(char *a) {  
    ...  
}  
void p2(char *b) {  
    char c;  
    XXX  
}
```

Cocher les valeurs de XXX qui correspondent à un appel possible.

- ☐ p1(&c);
☐ p1(c);
☐ p1(b);

- ☐ p1(&b);
☐ Aucune de ces réponses n'est correcte.

Question 27 Soit le programme suivant :

```
#include <stdio.h>  
int f1(int valeur) {  
    valeur = 0;  
    return valeur;  
}  
int main(){  
    int donnee = 20;  
    int donnee_retournee = f1(donnee); // la fonction utilise la valeur de donnee  
    printf("donnee : %i \n", donnee);  
    printf("donnee_retournee : %i \n", donnee_retournee);  
}
```

Quelles sont les valeurs de `donnee` et `donnee_retournee` à la fin du programme principal ?

- ☐ `donnee = 0` et `donnee_retournee = 20` ☐ `donnee = 0` et `donnee_retournee = 0`
☐ `donnee = 20` et `donnee_retournee = 20` ☐ `donnee = 20` et `donnee_retournee = 0`

Question 28 Soit le programme suivant :

```
#include <stdio.h>  
int f1(int* valeur) {  
    *valeur += *valeur;  
    return *valeur;  
}  
  
int main(){  
    int donnee = 20;  
    int donnee_retournee = f1(&donnee);  
    printf("donnee : %i \n", donnee);  
    printf("donnee_retournee : %i \n", donnee_retournee);  
}
```

Quelles sont les valeurs de `donnee` et `donnee_retournee` à la fin du programme principal ?

- ☐ `donnee = 40` et `donnee_retournee = 40` ☐ `donnee = 0` et `donnee_retournee = 20`
☐ `donnee = 20` et `donnee_retournee = 40` ☐ `donnee = 20` et `donnee_retournee = 20`