



Nom : Prénom :

Groupe :


Langage de programmation

Interrogation de novembre 2023

1. Connaissance du vocabulaire et compréhension des concepts généraux				
TRÈS INSUFFISANT → max 6/20	INSUFFISANT -1	JUSTE SUFFISANT	SATISFAISANT +1	BIEN ACQUIS +5
Maîtriser les principes généraux et le vocabulaire de la programmation.				
Commentaires :				
2. Utilisation des notions de base du C				
TRÈS INSUFFISANT → max 6/20	INSUFFISANT -2	JUSTE SUFFISANT	SATISFAISANT +2	BIEN ACQUIS +5
Comprendre et utiliser à bon escient les variables, opérateurs, expressions, instructions et le contrôle de flux. Comprendre, critiquer et corriger un programme simple.				
Commentaires : Question 1 (c) : la variable devrait être de type booléen Question 2 : les valeurs sont mauvaises. Le résultat final devrait être : nbEtudiants = 29, nbReussites = 9, coteMoyenne = 14.0) Question 3 (b) : le while devrait être un for Question 4 (c) : pas besoin d'affectation Question 5 (c) : pas besoin du else Question 6 : pas besoin du if				
3. Utilisation des notions avancées du C				
TRÈS INSUFFISANT → max 6/20	INSUFFISANT -2	JUSTE SUFFISANT	SATISFAISANT +2	BIEN ACQUIS +4
Comprendre et utiliser à bon escient les fonctions et les tableaux à simple indice. Comprendre, critiquer et corriger un programme simple.				
Commentaires :				
4. Algorithmique				
TRÈS INSUFFISANT → max 6/20	INSUFFISANT pénalité	JUSTE SUFFISANT		
Concevoir des algorithmes simples à partir de spécifications.				
Commentaires :				
5. Clean code et qualité du code				
TRÈS INSUFFISANT → max 6/20	INSUFFISANT pénalité	JUSTE SUFFISANT		
Appliquer des règles de clean code et des conventions.				
Commentaires : Question 5 (b) : mal indenté				

Note finale : 10/20

Consignes

- La durée prévue est de **2 heures**.
- Concernant le document Word reçu, vous devez
 - remplir la déclaration sur l'honneur qui se trouve au début du questionnaire, sous peine de rendre votre copie invalide ;
 - renommer ce document Word en **LP_XX_BG_Nom_Prénom.docx** où **B** doit être le numéro de votre bloc (1 ou 2) et **G** la lettre de votre groupe, Nom et Prénom sont remplacés par votre nom et votre prénom, et **XX** correspond aux initiales de votre section (DA/IA/IR/RT/TI) ;
 - répondre sous l'énoncé de chaque question par le pseudo-code (diagramme d'action) ou le code en langage C ou une explication en **français**, selon la question ;
 - veiller à la lisibilité de votre code : il doit être en **police monospace** (Consolas ou Courier, par exemple), il doit être en **noir sur fond bleu clair** comme dans les cadres réservés aux réponses...
 - **N'hésitez pas à coder dans votre éditeur de code et à copier le résultat pour le coller dans les cadres grâce à l'option  du clic droit de la souris.**
 - gérer la mise en page pour que la lecture en soit facilitée ; entre autres :
 - éviter les sauts de page mal placés (comme au milieu d'un code),
 - éviter les lignes de code trop longues qui "reviennent au début",
 - indenter correctement votre code
- **Au plus tard à la fin du temps imparti**, vous soumettez votre copie complétée et renommée dans le Devoir prévu à cet effet sur Moodle.
- Si vous rencontrez un problème ou si vous avez une question à poser, faites appel directement au professeur de LP présent dans le local.
Pour les DA/IA, des précisions destinées à tous pourront être postées par un enseignant dans le canal du cours de LP (#langage-c) sur **Discord**. Vous devez donc être joignable sur **Discord** durant cette période.
- Le type de question, les critères d'évaluation et de réussite sont ceux de la grille d'évaluation reprise au début du questionnaire.
- Vous pouvez bien sûr disposer de toutes vos notes de cours, de résumés, d'exercices résolus, de recherches internet... à l'exception de "l'appel à un ami".
Si plusieurs étudiants rendent des copies identiques, en tout ou en partie, chacun d'eux se verra attribuer un **zéro** pour l'ensemble de l'épreuve.

Question 1

Écrivez, en langage C, chacune des instructions correspondants aux descriptions suivantes.
Attention aux noms de variables, aux types, aux opérateurs... utilisés.

- (a) Déclarez et initialisez une variable permettant de mémoriser une date au format AAAAMMJJ.
La valeur initiale est votre date de naissance.

Le tout est à réaliser en une seule instruction.

```
Int dateNaissance = 20030616 ;
```

- (b) Déclarez et initialisez une variable permettant de mémoriser le numéro d'un mois (1 pour janvier, 2 pour février...).
- La valeur initiale doit correspondre au numéro du mois de la date déclarée au point (a).
Utilisez l'expression adéquate (n'utilisez pas de littéral).

Le tout est à réaliser en une seule instruction.

```
int moisNaissance = (dateNaissance % 10000) / 100;
```

- (c) Déclarez et initialisez une variable de façon à y mémoriser le fait que le numéro du mois correspond à un mois du premier semestre (inférieure ou égale à 6), ou pas.
Utilisez l'expression adéquate (n'utilisez pas de littéral).

Le tout est à réaliser en une seule instruction.

```
int estPremierSemestre = (moisNaissance <= 6) ? 1 : 0;  
bool estPremierSemestre = moisNaissance <= 6 ;
```

Question 2

Sur base des informations ci-dessous, complétez le tableau avec les valeurs des variables suite aux instructions consécutives suivantes.

Nom de la variable	nbEtudiants	nbReussites	sommeCotes	coteMoyenne
Type de la variable	int	int	int	double
Valeur initiale	22	-	421	-
nbEtudiants += 7;	29	9 rien	421	14 rien
nbReussites = nbEtudiants / 3;	36 29	12 9	421	11 rien
coteMoyenne = sommeCotes / nbEtudiants;	43 29	14 9	421	9 14

Question 3

Traduire les parties de DA suivantes en langage C, sans oublier de déclarer les variables et en utilisant les structures de contrôle les plus adéquates. Les expressions sont exprimées en français, pensez à les traduire correctement en C !

Notes :

- pas besoin de la fonction principale,
- pas besoin des bibliothèques,

(a)

```
nombre = 42
if (nombre est divisible par 5)
    sortir nombre, " est divisible par 5"
sinon
    sortir nombre, " n'est pas divisible par 5"
```

```
#include <stdio.h>

void main(void) {
    int nombre = 42;
    if (nombre % 5 == 0)
    {
        printf("%d\n est divisible par 5", nombre);
    } else
    {
        printf("%d\n n'est pas divisible par 5", nombre);
    }
}
```

(b)

```
nbJoueursScore500 = 0
iJoueur = 1
while (iJoueur est inférieur ou égal à 100)
    obtenir score
    if (score >= 500)
        nbJoueursScore500++
incrémenter iJoueur
```

```
#include <stdio.h>

void main(void) {
    int nbJoueursScore500 = 0;
    int iJoueur = 1;

    while (iJoueur <= 100) {
        int score = 0;
        printf("Introduisez le score :\n", score);
        scanf_s("%d", &score);

        if (score >= 500)
```

```

        {
            nbJoueursScore500++;
        }

        iJoueur++;
    }
}

```

CODE DU HAUT CORRECT MAIS PAS LA STRUCTURE LA PLUS ADEQUATE. LA STRUCTURE LA PLUS ADEQUATE EST C'ELLE DU BAS (FOR).

```

int nbJoueursScore500 = 0 ;

for (int i = 0 ; i <= 100 ; i++){
    int score ;
    printf("Entrez le score du joueur:");
    scanf_s("%d", &score) ;
    if(score >= 500){
        nbJoueursScore500++ ;
    }
}

```

Question 4

Un programmeur a écrit la ligne de code suivante :

```
nbInscrits = ++nbInscrits;
```

- Si `nbInscrits` vaut 42 avant la ligne de code ci-dessus, quelle est sa valeur après ?

43

- Pensez-vous qu'elle est bien écrite (clean code) ? Justifiez votre réponse en expliquant, avec les mots adéquats, la suite des opérations effectuées, sur base de la valeur initiale 42.

Elle incrémente d'abord `nbInscrits` de 1, puis réaffecte cette nouvelle valeur à `nbInscrits`.

Réponse du haut bonne mais meilleur réponse :

Non, il s'agit d'une préincrémentation. D'abord la valeur de `nbInscrits` est augmentée de 1, ensuite elle est affectée à la variable `nbInscrits`. Il y a donc 2 étapes alors que seule la première est nécessaire.

- Si vous pensez qu'il y a une « meilleure » façon de faire, proposez-la ci-dessous.

```
nbInscrits = nbInscrits++ ;
nbInscrits++ ;
```

Question 5

Dans le cadre d'une enquête auprès des étudiants de l'Henallux, on désire connaître la raison pour laquelle ils préfèrent les cours en présentiel à ceux donnés à distance via Teams. Pour cela, une question leur est soumise avec plusieurs raisons dont une seule (la principale) peut être choisie. Les propositions sont les suivantes :

1. Je n'ai pas d'ordinateur personnel
2. J'ai une mauvaise connexion internet
3. Je n'ai pas de pièce où m'isoler
4. J'ai besoin de voir mes copains
5. Je m'endors devant l'écran
6. J'ai tendance à préférer jouer

- (a) Déclarez la variable `numRaison` qui permet de mémoriser le numéro correspondant à la raison choisie (Exemple : « Je n'ai pas d'ordinateur personnel » → 1).

```
Int numRaison;
```

- (b) L'étudiant doit entrer un numéro de raison valide...

Écrivez les instructions qui lui permettent d'obtenir un numéro, qui l'avertissent d'une éventuelle erreur et lui permettent de rentrer un nouveau numéro de raison si le précédent n'est pas correct, et ce, jusqu'à ce que le numéro de raison soit correct.

```
#include <stdio.h>

void main(void){
    int numRaison;
    int raisonMax = 6;

    printf("Introduisez un numero de raison valide entre 1 et 6 :\n");
    scanf_s("%d", &numRaison);

    while (numRaison < 1 || numRaison > raisonMax) {
        printf("Saissi invalide, introduisez un numero entre 1 et 6 :\n");
        scanf_s("%d", &numRaison);
    }
    printf("Vous avez choisi la raison numero %d\n", numRaison);
    printf("Vous avez choisi la raison numero %d\n", numRaison); (correction indentation)
}
```

- (c) Les raisons sont catégorisées comme suit : les raisons 1 à 3 sont « technique » et les raisons 4 à 6 sont « humaine ». On propose le code suivant pour afficher la catégorie correspondant à la raison choisie.

Sachant que le numéro de raison est valide (compris entre 1 et 6 inclus), si vous estimez qu'il y a des erreurs de syntaxe ou des améliorations à apporter (algorithmique et/ou clean code), écrivez le code modifié ci-dessous.

```
if (1 < numRaison < 3) {
    printf("Categorie technique");
} else {
    if (4 < numRaison < 6) {
        printf("Categorie humaine");
    }
}
```

```
void main(void) {  
    if (numRaison >= 1 && numRaison <= 3) {  
        printf("Categorie technique");  
    } else if (numRaison >= 4 && numRaison <= 6) {  
        printf("Categorie humaine");  
    }  
}  
  
if (1 < numRaison < 3) {  
    printf("Categorie technique");  
} else {  
    printf("Categorie humaine");  
}
```

Question 6

Écrivez le programme qui permet sur base d'une valeur obtenue de l'utilisateur et mémorisée dans nb, d'afficher le triangle suivant :

Si nb vaut 3 :

```
+++  
++  
+
```

Si nb vaut 5 :

```
+++++  
++++  
+++  
++  
+
```

Notes :

- [Précondition] On supposera que le nombre obtenu est un nombre entier > 0 et < 10 .
- Utilisez les instructions les plus adéquates (comme d'habitude...).
- N'oubliez pas les espaces et les passages à la ligne...
- Ce n'est pas grave si on affiche un espace de trop en fin de ligne.

```
#include <stdio.h>  
  
void main(void) {  
    int nb;  
  
    printf("Entrez un nombre (entre 1 et 9) : ");  
    scanf("%d", &nb);  
  
    if (nb > 0 && nb < 10) {  
        for (int ligne = nb; ligne > 0; ligne--) {  
            for (int colonne = 0; colonne < ligne; colonne++) {  
                printf("+ ");  
            }  
            printf("\n");  
        }  
    } else {  
        printf("Le nombre doit etre compris entre 1 et 9.\n");  
    }  
}
```

CODE CORRECT MAIS ILS FAUT ENLEVER LE IF ET FOR CAR PRECONDITION DANS L'ENONCE.

```
#include <stdio.h>  
  
void main(void) {  
    int nb;  
  
    printf("Entrez un nombre (entre 1 et 9) : ");  
    scanf("%d", &nb);  
  
    for (int ligne = nb; ligne > 0; ligne--) {
```



```
        for (int colonne = 0; colonne < ligne; colonne++) {  
            printf("+ ");  
        }  
        printf("\n");  
    }  
}
```