

Nom, Prénom :

Travail écrit Info1-TIN-B  
TE-01*Haute École d'Ingénierie et de Gestion du Canton de Vaud**Département TIN*

Prof. Yves Chevallier

4 novembre 2025

Question	Points	Score
1. Numération	10	
2. Syntaxe	10	
3. Entrées & sorties	10	
4. Structures de contrôle	10	
5. Programmation	10	
Total :	50	

**Consignes :**

- Écrire votre **nom** et votre **prénom** sur la première page.
- Écrire **lisiblement**, au stylo ou au crayon à papier gras.
- Répondre aux questions dans les espaces appropriés.
- Relire toutes vos réponses avant de rendre votre travail.
- Rendre toutes les feuilles de ce travail écrit.
- Les réponses données sur les feuilles de brouillon ne sont **pas acceptées**.
- Aucun moyen de communication autorisé.
- Toutes les réponses concernent le langage C et son standard C20.

1. Pour chacune des constantes littérales suivantes indiquez leur équivalent :

1. binaire,
2. octal,
3. décimal signé,
4. décimal non signé et
5. hexadécimal.

Considérez que les nombres sont stockés en mémoire sur **8 bits** et que la représentation signée utilise le **complément à deux**. Complétez le tableau ci-dessous en remplissant les cases vides.

bin	octal	int	uint	hex
0b01011010				
			199	
		-112		
	0175			
				0xfa

2. (a) Pour chacun des identificateurs suivants, indiquez s'ils sont corrects selon le standard C, et si non, proposez un nom compatible en C.

i. résumé

i. \_\_\_\_\_

ii. \_AMountain\_

ii. \_\_\_\_\_

iii. while

iii. \_\_\_\_\_

iv. std99

iv. \_\_\_\_\_

v. data-stream

v. \_\_\_\_\_

- (b) Pour chaque constante littérale suivante, indiquez si elle est **correcte** et, si applicable, précisez le **type** associé.

i. 42uL

i. \_\_\_\_\_

ii. 0758

ii. \_\_\_\_\_

iii. .128f

iii. \_\_\_\_\_

iv. '\x41'

iv. \_\_\_\_\_

v. 0b00001010

v. \_\_\_\_\_

3. (a) Pour les appels de fonction `printf` suivants, indiquez l'affichage exact produit sur *stdout* ou, en cas d'erreur, la nature de cette dernière. Représentez la sortie dans l'espace approprié, un caractère par case. Utilisez la notation des caractères d'échappement du langage C pour les caractères non imprimables (p. ex. `\n`). Terminez chaque sortie par une croix sur toute la case.

Considérez les états des variables suivants :

```
short s = -27;
unsigned char uc = 201;
char c = 'c'; // Valeur ASCII de 'c': 99
double x = 2.71828;
```

- i. `printf("|0x%05x|\n", uc);`

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- ii. `printf("%+7.2f\n", s / 2.0);`

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- iii. `printf("%c%c%c%c%hhd", c, c - 2, 'a' + 2, 97, c);`

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- iv. `printf(">%-06.1f<", x);`

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- v. `printf("%-*.*f%d\n", 7, 3, 5.4321, 42);`

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

(b) Entrée standard.

Soient les déclarations suivantes :

```
int r = 0, n = 0, m = 0;  
double y = 0.0;  
char ch = '0';
```

Pour les appels `sscanf` qui suivent, indiquez :

1. la valeur des variables affectées ;
2. la valeur de retour `r`.

i. `r = sscanf("_-15x1f", "%d%i", &n, &m);`

i. \_\_\_\_\_

ii. `r = sscanf("42kg", "%2d%c", &n, &ch);`

ii. \_\_\_\_\_

iii. `r = sscanf("0.75,12", "%lf,%d", &y, &n);`

iii. \_\_\_\_\_

iv. `r = sscanf("abc", "%d", &n);`

iv. \_\_\_\_\_

v. `r = sscanf("9.82", "%d%d", &n, &m);`

v. \_\_\_\_\_

4. Donnez les valeurs affichées sur *stdout* lors des séquences suivantes :

(a) 

```
int s = 1;
while (s < 20) {
    printf("%d ", s);
    s <<= 1;
}
```

(a) \_\_\_\_\_

(b) 

```
for (int i = 5; i > 0; --i) {
    if (i % 2 == 0) continue;
    printf("%d", i);
}
```

(b) \_\_\_\_\_

(c) 

```
int total = 0;
for (int i = 1; i <= 3; ++i) {
    for (int j = i; j <= 3; ++j) {
        total += j;
    }
}
printf("%d", total);
```

(c) \_\_\_\_\_

(d) 

```
for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) {
        if (i == j) break;
        printf("%d%d ", i, j);
    }
}
```

(d) \_\_\_\_\_

(e) 

```
int i = 1, t = 0;
do {
    t += i;
    printf("%d;", t);
    i += 2;
} while (t < 15);
```

(e) \_\_\_\_\_



