

Première partie, questions à choix multiple

Pour chaque question, marquer la case correspondant à la réponse correcte sans faire de ratures. Il n'y a qu'une seule réponse correcte par question.

Question [SCQ-01] :

On considère l'algorithme suivant :

algo
entrée : L liste de n lettres
sortie : valeur binaire (oui/non)
<pre> s ← oui Pour i allant de 1 à $\lfloor \frac{n}{2} \rfloor$ Si $L(i) \neq L(n+1-i)$ s ← non Sortir: s </pre>

Pour quelle liste de lettres en entrée la sortie de l'algorithme ci-dessus est-elle un oui?

- MALAYALAM
 CHACHACHA
 HAHAHAH
 ALABAMA

Question [SCQ-02] :

On considère l'algorithme suivant :

algo
entrée : n nombre entier positif
sortie : nombre entier positif
<pre> s ← 0 Pour i allant de 1 à n Pour j allant de 1 à i Pour k allant de j à i s ← s + 1 Sortir: s </pre>

Quelle est la sortie de **algo**(3) ?

- 10
 9
 6
 27

Question [SCQ-03] :

Quelle est la complexité temporelle de l'algorithme de la question 2 ?

- $\Theta(n)$
 $\Theta(n^2)$
 $\Theta(n^3)$
 $\Theta(n^4)$

Question [SCQ-04] :

On considère l'algorithme suivant :

algo
entrée : A, B deux nombres entiers positifs sortie : nombre entier positif
<pre> Si $B = 0$ Sortir: 1 Si B est pair Sortir: $(\text{algo}(A, B/2))^2$ Si non Sortir: $(\text{algo}(A, (B - 1)/2))^2 \cdot A$ </pre>

Quelle est la sortie de $\text{algo}(A, B)$?

- A^B

 $A \cdot B$

 B^A

 A^2 si B est pair;

 A^3 si B est impair

Question [SCQ-05] :

Si A, B sont chacun des nombres à n chiffres, combien d'instructions seront-elles lues par l'algorithme de la question 4 jusqu'à ce qu'il termine ? (ceci correspond à la complexité temporelle de l'algorithme, si on ne tient pas compte de la complexité des opérations en elles-mêmes)

- $\Theta(n)$

 $\Theta(10^n)$

 $\Theta(2^n)$

 $\Theta(n^n)$

Question [SCQ-06] :

On considère deux algorithmes $\text{algo1}(n)$ et $\text{algo2}(n)$ dont les complexités temporelles sont respectivement données par $\frac{n}{2}$ et $2n^2$ (on suppose ici que ces complexités temporelles sont *exactes*).

L'exécution de $\text{algo1}(1'000)$ sur une machine donnée (disons la machine A) termine en 1 minute.

L'exécution de $\text{algo2}(1'000)$ sur une autre machine (disons la machine B) termine également en 1 minute.

Si on exécute maintenant $\text{algo1}(10'000)$ sur la machine A et $\text{algo2}(10'000)$ sur la machine B, alors :

- $\text{algo1}(10'000)$ termine en 10 minutes et $\text{algo2}(10'000)$ termine en 100 minutes
 $\text{algo1}(10'000)$ termine en 5 minutes et $\text{algo2}(10'000)$ termine en 200 minutes
 $\text{algo1}(10'000)$ termine en 10 minutes et $\text{algo2}(10'000)$ termine en 200 minutes
 $\text{algo1}(10'000)$ termine en 5 minutes et $\text{algo2}(10'000)$ termine en 100 minutes

Question [SCQ-07] :

Soient $x = 108$ et $y = 124$ deux nombres entiers positifs, chacun représenté en binaire avec la représentation des nombres entiers positifs sur 8 bits.

On effectue une opération XOR bit-à-bit sur ces deux nombres (c'est-à-dire qu'on effectue l'opération XOR sur le premier bit de x et le premier bit de y , puis la même opération sur le second bit de x et le second bit de y , etc., sans jamais considérer de retenue). On obtient ainsi une nouvelle séquence de 8 bits. Quel est le nombre entier positif z représenté par cette nouvelle séquence de 8 bits ?

 16 232 8 222**Question [SCQ-08] :**

On considère le problème suivant :

Etant donné une liste L de n nombres entiers relatifs, identifier le plus grand sous-ensemble $S \subset \{1, \dots, n\}$ (i.e., le sous-ensemble avec le plus grand nombre d'éléments) tel que $\sum_{i \in S} L(i) \geq 0$.

Parmi les affirmations ci-dessous, laquelle est vraie?

 Ce problème fait partie de la classe P. Ce problème fait partie de la classe NP, mais on ne sait pas s'il fait partie de la classe P. Ce problème fait partie de la classe NP, donc il ne fait pas partie de la classe P. On ne sait pas si ce problème fait partie de la classe P, ni s'il fait partie de la classe NP.**Question [SCQ-09] :** Soient trois variables définies comme suit:

```
int u = 12, v[100];
```

```
float score;
```

Plus loin dans le même bloc il y a un appel de fonction:

```
score = calculer(u, v);
```

Quelles sont toutes les déclarations possibles, i.e., qui ne déclenchent pas d'erreur de compilation, pour la fonction `calculer` parmi les déclarations suivantes?

(a) `int calculer(long a, int* b);`(b) `float calculer(int a, int& b);`(c) `double calculer(int a, int b[]);`(d) `float calculer(long a, int** b);` (a) et (c) (a) et (d) (b) et (c) Uniquement (c)

CATALOGUE

Question [SCQ-10] : Qu'affiche ce code?

```
int tab[3] = {1, 2, 3};
tab[++tab[0]]--;
printf("%d %d %d\n", tab[0], tab[1], tab[2]);
```

2 2 2

2 1 3

Il ne compile pas.

1 2 3

Question [SCQ-11] : Qu'affiche le code suivant?

```
int a[] = {1, 2, 3, 4, 5};
for (int* b = a + 4; b >= a; b--)
{
    printf("%d ", b[0]);
}
```

5 4 3 2 1

9 8 7 6 5

5 5 5 5 5

Il ne compile pas.

Question [SCQ-12] : Qu'affiche ce code?

```
#define T 3
void f(const char *txt, char *out, int b, int u, int t)
{
    out[u] = txt[b];
    if (txt[b] == '\0')
    {
        return;
    }
    if (txt[b] == 'u' && t > 0)
    {
        f(txt, out, b, u + 1, t - 1);
    }
    else
    {
        f(txt, out, b + 1, u + 1, T);
    }
}

int main()
{
    char string[10];
    f("but", string, 0, 0, T);
    printf("%s\n", string);
}
```

buuuut

Ne termine pas.

tub

but

CATALOGUE

On définit les structs suivantes:

```
typedef struct _point
{
    double x, y;
} point_t;
```

```
typedef struct _polygone
{
    int n;
    point_t *sommets;
} polygone_t;
```

Dans la fonction main() on définit les variables suivantes:

```
point_t carré[] =
{
    {0, 0}, {0, 5}, {5, 5}, {5, 0}
};
```

```
polygone_t p;
p.n = 4;
p.sommets = carré;
```

```
polygone_t *copie1 = &p;
```

```
polygone_t *copie2 = malloc(sizeof(polygone_t));
copie2->n = p.n;
copie2->sommets = p.sommets;
```

```
polygone_t *copie3 = malloc(sizeof(polygone_t));
copie3->n = p.n;
copie3->sommets = malloc(p.n * sizeof(point_t));
for (int i=0; i<p.n; i++)
{
    copie3->sommets[i] = p.sommets[i];
}
```

Question [SCQ-13] : Le code suivant se trouve dans main() juste après les instructions ci-dessus. Qu'affiche-t-il?

```
p.n = 3;
printf("%d %d %d\n", copie1->n, copie2->n, copie3->n);
```

3 4 4

4 4 4

3 3 4

3 3 3

Question [SCQ-14] : La fonction `double perimetre(polygone_t *poly)` calcule le périmètre d'un polygone. Qu'affiche le code suivant? Il se trouve dans main() juste après le code de la question précédente.

```
p.n = 4;
for (int i = 0; i < 4; i++)
{
    carré[i].x *= 2;
    carré[i].y *= 2;
}
printf("%.1lf %.1lf %.1lf\n", perimetre(copie1), perimetre(copie2), perimetre(copie3));
```

40.0 40.0 20.0

40.0 40.0 40.0

40.0 20.0 20.0

20.0 20.0 20.0

CATALOGUE

Question [SCQ-15] : Voici quelques définitions de fonctions:

```
void swap1(int u, int v)
{
    int aux = u;
    u = v;
    v = aux;
}
```

```
void swap2(int *u, int *v)
{
    int *aux = u;
    u = v;
    v = aux;
}
```

```
void swap3(int *u, int *v)
{
    int aux = u[0];
    u[0] = v[0];
    v[0] = aux;
}
```

```
void swap4(int *u, int *v)
{
    int **aux = &u;
    &u = &v;
    &v = aux;
}
```

On définit les variables `int a = 1, b = 2`; Quel appel échange leur contenu?

`swap3(&a, &b);` `swap1(a, b);` `swap2(&a, &b);` `swap4(&a, &b);`

Question [SCQ-16] : Qu'affiche le code suivant?

```
char *tag = "My name is xxx", *bob = "Robert", result[100], *r = result;
```

```
for (char *p = tag, *q = bob; *p != '\0' && *q != '\0'; p++)
```

```
{
    if (*p == 'x')
    {
        *r = *q;
        q++;
    }
    else
    {
        *r = *p;
    }
    r++;
}
```

```
*r = '\0';
```

```
printf("%s\n", result);
```

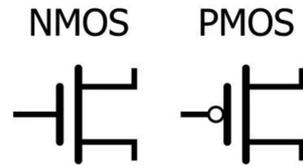
My name is Rob My name is Bob My name is Robert My name is
Rob2edmsaere2323

Question 18: *Cette question est notée sur 4 points.*

0 1 2 3 4

Réservé au correcteur

En utilisant les transistors n-mos et p-mos vus au cours:



construire une porte NOR (=NON OU) dont la sortie s vaut 1 si et seulement si les deux entrées x et y valent chacune 0.

Rappel: Un transistor n-mos laisse passer le courant entre l'émetteur et le collecteur si et seulement si la tension d'entrée à la base est haute; il se passe exactement le contraire dans un transistor p-mos.



CATALOGUE

On définit les deux struct suivantes:

```
typedef struct _cell
{
    char *mot;
    struct _cell *next;
} cell_t;
```

```
typedef struct _liste
{
    cell_t *premier;
    cell_t *dernier;
} liste_t;
```

c) (3 points) Écrivez une fonction qui reçoit une liste et un pointeur vers une chaîne de caractères contenant un seul mot. Cette fonction rajoute une nouvelle cellule ayant pour contenu le pointeur **un_mot à la fin** de la liste chaînée `liste`. Cette fonction doit mettre à jour les champs `premier` et `dernier`. Pour une liste vide ces champs auront la valeur `NULL`.

```
void enfiler_mot(liste_t *liste, char* un_mot) {
```



```
}
```

