

**Question 7 :**

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 ?

 222 16 8 232**Question 8 :**

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 9 :** 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 (d) Uniquement (c) (a) et (c) (b) et (c)



**Question 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 Il ne compile pas. 1 2 3 2 1 3

**Question 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]);
}
```

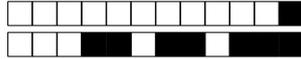
 Il ne compile pas. 5 5 5 5 5 5 4 3 2 1 9 8 7 6 5

**Question 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);
}
```

 tub but Ne termine pas. buuuut



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 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 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 20.0 20.0 20.0 40.0 20.0 20.0



**Question 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 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 Robert       My name is Bob       My name is  
Rob2edmsaere2323       My name is Rob