
ICC-C Programming

Question P1 (2 points) : Qu'est-ce que ce programme affiche ?

```
int a = 1, b = 11, c = 111;
a += c / (b + a);
printf("%d %d %d\n", (a - b) >= (b - c), a, a == b);
```

(A)

(B)

(C)

(D)

Question P2 (2 points) : Qu'est-ce que ce programme affiche ?

```
int m = 2;
int cnt = 0;

for (int i = 1; i <= 100; i++){
    if (i % m) cnt++;
    m++;
}

printf("%d", cnt);
```

(A)

(B)

(C)

(D)

Question P3 (2 points) : Qu'est-ce que ce programme affiche ?

```
int n = 2048;
int cnt = 0;

do {
    n = n / 2;
    cnt++;
} while (n > 1);

printf("%d", cnt);
```

(A)

(B)

(C)

(D)

.....

Question P4 (2 points) : Qu'est-ce que ce programme affiche ?

```
#include <stdio.h>
int main()
{
    int var1 = 2, var2 = -3;

    if (var2 > 0 || var1 > 0) {
        var2--;
        var1++;
    }

    if (var2 < 0 || var1 < 0) {
        var2 += 2;
        var1 += 2;
    }

    if (var1 && var2)
        printf("%d ", var1 * var2);
    else
        printf("%d ", -var1 * var2);

    return 0;
}
```

(A)

(B)

(C)

(D)

Question P5 (2 points) : Qu'est-ce que ce programme affiche ?

```
#include <stdio.h>
#include <string.h>
int main()
{
    char line[] = "La vie est belle.";
    char c;
    int cnt = 0;

    for (int i = 0; i < strlen(line); i++){
        c = line[i];
        switch (c) {
            case 'L':
            case 'l': cnt++; break;
            default: break;
        }
    }

    printf("%d", cnt);
    return 0;
}
```

(A)

(B)

(C)

(D)

.....

Question P6 (2.5 points) : Qu'est-ce que ce programme affiche ?

```
#include <stdio.h>
#define N 10
int main() {
    int a[N] = {9, 38, 7, 31, 50, 2, 22, 27, 3, 41};
    int i = 0, cnt = 0;
    while (i < N) {
        i++;
        if (a[i] > N) continue;
        cnt += a[i];
    }
    printf("%d", cnt);
    return 0;
}
```

Question P7 (2.5 points) : Qu'est-ce que ce programme affiche ?

```
#include <stdio.h>
#define N 10

void make_it_double(int* n){
    int res = *n * 2;
    *n = res;
}

void make_it_triple(int n) {
    int res = n * 3;
    n = res;
}

int main() {
    int a[N] = {9, 38, 7, 31, 50, 2, 22, 27, 3, 41};
    for (int i = 0; i < N; i++){
        if (i % 2) make_it_double(&a[i]);
        else make_it_triple(a[i]);
    }

    printf("%d", a[0] + a[N-1]);
    return 0;
}
```

.....

Question P8 (2.5 points) : Combien d'étoiles ce programme affiche ?

```
#include <stdio.h>

void print_stars(int n){
    if (n < 0) return;
    else if (n < 3) {
        printf("*");
        print_stars(n-1);
    }
    else {
        printf(" ** ");
        print_stars(n-2);
    }
}

int main() {
    print_stars(10);
    return 0;
}
```

Question P9 (2.5 points) : Qu'est-ce que ce programme affiche ?

```
#include <stdio.h>
#define N 10

int main() {
    int a[N] = {9, 38, 7, 31, 50, 2, 22, 27, 3, 41};
    int b[N] = {0};

    for (int i = 1; i < N - 1; i++) {
        *(b + i) = *(a + i + 1) + *(a + i - 1) - *(a + i);
    }
    printf("%d", *(b + N/2));
    return 0;
}
```

.....

Question P10 (2.5 points) : Qu'est-ce que ce programme affiche ?

```
#include <stdio.h>

int main() {
    int x[2] = {1, 2};
    int y[2] = {3, 4};
    int z[2] = {5, 6};
    int *p1, *p2;

    p1 = x;
    p2 = z + 1;

    *(p1 + 1) += *p2;
    p2 -= 1;

    p1 = y;
    *(p1 + 1) = *p2;

    *p2 = *p1;

    printf("%d", x[1] + y[1] + z[1]);

    return 0;
}
```

Question P11 (2.5 points) : Qu'est-ce que ce programme affiche ?

```
#include <stdio.h>
#include <string.h>

int main() {
    char s[100] = "Bond. James Bond.";
    char t[100] = {'\0'};
    char* key = "Bond.";
    int cnt = 0;
    char *cp = strstr(s, key);

    while (cp != NULL) {
        strcat(t, key);
        cnt++;
        cp = strstr(s + cnt * strlen(key), key);
    }

    printf("%d", cnt);
    return 0;
}
```

.....

Question P12 (5 points) : Écrivez la fonction `count_digits` qui attend deux arguments:

- un entier positif `n` et
- un tableau `d` de dix éléments de type `int`, tous initialisés à zéro.

Cette fonction doit compter le nombre de fois qu'un chiffre (0, 1, 2, ..., 9) apparaît dans le nombre `n` et remplir le tableau `d` en conséquence. L'élément du tableau `d` à l'indice `i` correspond au nombre d'occurrences du chiffre `i`. Pour un `n` négatif ou égal à zéro, la fonction ne doit apporter aucune modification au tableau `d`.

Exemples:

(1) `count_digits(155027,d)` aura comme effet :

indice <i>i</i>	0	1	2	3	4	5	6	7	8	9
<code>d[i]</code>	1	1	1	0	0	2	0	1	0	0

(2) `count_digits(8285260,d)` aura comme effet :

indice <i>i</i>	0	1	2	3	4	5	6	7	8	9
<code>d[i]</code>	1	0	2	0	0	1	1	0	2	0

(3) `count_digits(1575559,d)` aura comme effet :

indice <i>i</i>	0	1	2	3	4	5	6	7	8	9
<code>d[i]</code>	0	1	0	0	0	4	0	1	0	1

Question P13 (5 points) : Écrivez une fonction `pearson_corr` qui attend deux arguments:

- `x`, un tableau d'éléments de type `double`,
- `y`, un tableau d'éléments de type `double` et
- `n`, le nombre d'éléments dans chacun des tableaux `x` et `y`.

Votre fonction doit retourner le coefficient de corrélation de Pearson r , qui se calcule à l'aide de formule suivante :

$$r = \frac{\sum_{i=1}^n ((x_i - \bar{x}) \cdot (y_i - \bar{y}))}{\sqrt{(\sum_{i=1}^n (x_i - \bar{x})^2) \cdot (\sum_{i=1}^n (y_i - \bar{y})^2)}} \quad (1)$$

où $\bar{x}$ et $\bar{y}$ sont les moyennes arithmétiques du tableau `x` et `y`, respectivement:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (2)$$