Quiz

Programmation impérative C

2025-2026

Variables

Quelle est la valeur initiale de a ?

```
int a;
```

A Sans initialisation explicite de la variable, sa valeur est indéterminée

B La variable a vaut 0, car la valeur par défaut du type int est 0

#QDLE#Q#A*B#25#

Quelle est la valeur initiale de a ?

```
int a;
```

A Sans initialisation explicite de la variable, sa valeur est indéterminée

B La variable a vaut 0, car la valeur par défaut du type int est 0

Quelle est la taille (en octets) de a ?

```
int a = 10;
```

A 2 octets

B 4 octets

C 8 octets

D Indéterminable sans connaître la plate-forme d'exécution

#QDLE#Q#ABCD*#30#

Quelle est la taille (en octets) de a ?

```
int a = 10;
```

A. 2 octets

B. 4 octets

C. 8 octets

D. Indéterminable sans connaître la plate-forme d'exécution

Quelle est la plage d'entiers que l'on peut coder avec une variable de type *char* ?

A [-128, 127]

B [0, 127]

C [0, 255]

D [0, 256]

#QDLE#Q#A*BCD#40#

Quelle est la plage d'entiers que l'on peut coder avec une variable de type *char* ?

A [-128, 127]

B [0, 127]

C [0, 255]

D [0, 256]

Quel type de données est le plus approprié pour stocker la valeur -1200 ?

A short int

B unsigned short int

C int

D long

#QDLE#Q#A*BCD#45#

Quel type de données est le plus approprié pour stocker la valeur -1200 ?

A short int

B unsigned short int

C int

D long

Quelle est la bibliothèque qui contient le prototypage de la fonction *printf* ?

A stdio.h

B stdlib.h

C strings.h

D Aucunes de ces réponses

#QDLE#Q#A*BCD#25#

Quelle est la bibliothèque qui contient le prototypage de la fonction *printf* ?

A stdio.h

B stdlib.h

C strings.h

D Aucunes de ces réponses

Qu'affiche le code suivant ?

```
double result = 1 / 3;
printf("%lf\n", result);
```

A 0.00000

B 0.33333

C 1/3

D Erreur à l'exécution

#QDLE#Q#A*BCD#40#

Qu'affiche le code suivant ?

```
double result = 1 / 3;
printf("%lf\n", result);
```

A 0.00000

B 0.33333

C 1/3

D Erreur à l'exécution

Qu'affiche le code suivant ?

```
int main(){
  float a = 0.1;
  if (a == 0.1)
      printf("equal\n");
  else
      printf("not equal\n");
  return 0;
}
```

A equal

B not equal

C On ne sait pas, ça dépend du système

#QDLE#Q#ABC*#45#

Qu'affiche le code suivant ?

```
int main(){
  float a = 0.1;
  if (a == 0.1)
      printf("equal\n");
  else
      printf("not equal\n");
  return 0;
}
```

A equal

B not equal

C On ne sait pas, ça dépend du système

Qu'affiche le code suivant ?

```
int main() {
  int a = 2-3;
  if (a)
      printf("OK\n");
  else
       printf("KO\n");
  return 0;
}
```

A OK

B KO

C Erreur à la compilation

D Erreur de segmentation

#QDLE#Q#A*BCD#40#

Qu'affiche le code suivant ?

```
int main() {
  int a = 2-3;
  if (a)
      printf("OK\n");
  else
       printf("KO\n");
  return 0;
}
```

A OK

B KO

C Erreur à la compilation

D Erreur de segmentation

Qu'affiche le code suivant ?

```
int main() {
  printf("Hello World! %d\n", x);
  return 0;
}
```

A Hello World! x;

B Hello World! suivi par une valeur aléatoire

C Hello World!

D Erreur à la compilation

#QDLE#Q#ABCD*#40#

Qu'affiche le code suivant ?

```
int main() {
  printf("Hello World! %d\n", x);
  return 0;
}
```

A Hello World! x;

B Hello World! suivi par une valeur aléatoire

C Hello World!

D Erreur à la compilation

Fonctions

Est-il possible de définir des valeurs par défaut pour les paramètres d'une fonction ?

```
void f(int num=0, int den=1);
```

A — Oui

B — Non

#QDLE#Q#AB*#20#

Est-il possible de définir des valeurs par défaut pour les paramètres d'une fonction ?

```
void f(int num=0, int den=1);
```

A. Oui

B. Non

Qu'affiche le code suivant ?

```
int test(int c) {
  printf("c\n");
  return c;
}

int main() {
  test(2+3);
  return 0;
}
```

A. 5

B. c

C. Erreur de segmentation

D. Erreur à la compilation

#QDLE#Q#AB*CD#45#

Qu'affiche le code suivant ?

```
int test(int c) {
  printf("c\n");
  return c;
}

int main() {
  test(2+3);
  return 0;
}
```

A 5

B c

C Erreur de segmentation

D Erreur à la compilation

Qu'affiche le code suivant ?

```
int func(int a) {
  a = 0;
  return a;
}
int main() {
   int a = 1;
   func(a);
   printf("%d\n",a);
   return 0;
}
```

#QDLE#Q#AB*CD#45#

A. 0

B. 1

C. Erreur de segmentation

D. Erreur à la compilation

Qu'affiche le code suivant ?

```
int func(int a) {
  a = 0;
  return a;
}
int main() {
   int a = 1;
   func(a);
   printf("%d\n",a);
   return 0;
}
```

A 0

B 1

C Erreur de segmentation

D Erreur à la compilation

Boucles et Tableaux

La boucle suivante est-elle syntaxiquement correcte ?

```
for(int i=0; i<10; i++) {
    printf("%d - ", i);
}
```

A Non, cette syntaxe n'est valable qu'en C++ (j'imagine)

B Oui, cette syntaxe (reprise de C++) existe depuis la version C ISO99

#QDLE#Q#AB*#25#

La boucle suivante est-elle syntaxiquement correcte ?

```
for(int i=0; i<10; i++) {
    printf("%d - ", i);
}
```

A Non, cette syntaxe n'est valable qu'en C++ (j'imagine)

B Oui, cette syntaxe (reprise de C++) existe depuis la version C ISO99

Est-ce que ce code compile sans erreur ?

```
int main() {
  int i;
  {
    int i=0;
  }
  return 0;
}
```

A Dépend de la norme C mise en œuvre par les compilateurs

B Oui

C Non

D Aucunes de ces réponses

#QDLE#Q#AB*CD#35#

Est-ce que ce code compile sans erreur ?

```
int main() {
  int i;
  {
    int i=0;
  }
  return 0;
}
```

A Dépend de la norme C mise en œuvre par les compilateurs

B Oui

C Non

D Aucunes de ces réponses

Qu'affiche le code suivant ?

```
int main () {
 int i = 0, j = 10;
 for(int i=0; i<=j; i++, j--)
     {};
 printf("%d %d", i, j);
 return 0;
}
```

A. 5 5

B. 6 4

C. 0 10

D. 0 4

#QDLE#Q#ABCD*#80#

Qu'affiche le code suivant ?

```
int main () {
 int i = 0, j = 10;
 for(int i=0; i<=j; i++, j--)
     {};
 printf("%d %d", i, j);
 return 0;
}
```

- **A** 5 5
- **B** 6 4
- **C** 0 10
- **D** 0 4

Qu'affiche le code suivant ?

```
int a = 3;
int b = 7;
int r = a++ + b++;
printf("%d-%d-%d", a, b, r);
```

- **A** 3-7-10
- **B** 3-7-12
- **C** 4-8-10
- **D** 4-8-12

#QDLE#Q#ABC*D#35#

Qu'affiche le code suivant ?

```
int a = 3;
int b = 7;
int r = a++ + b++;
printf("%d-%d-%d", a, b, r);
```

A 3-7-10

B 3-7-12

C 4-8-10

D 4-8-12

Qu'affiche le code suivant ?

```
int main() {
  a = 5;
  printf("%d\n", a++);
  return 0;
}
```

A 0

B 5

C 6

D Erreur à la compilation

#QDLE#Q#ABCD*#25#

Qu'affiche le code suivant ?

```
int main() {
  a = 5;
  printf("%d\n", a++);
  return 0;
}
```

- **A** 0
- **B** 5
- **C** 6
- **D** Erreur à la compilation

Qu'affiche le code suivant ?

```
int a = 3;
int b = 7;
int r = ++a + ++b;
printf("%d-%d-%d", a, b, r);
```

- **A** 3-7-10
- **B** 3-7-12
- **C** 4-8-10
- **D** 4-8-12

#QDLE#Q#ABCD*#30#

Qu'affiche le code suivant ?

```
int a = 3;
int b = 7;
int r = ++a + ++b;
printf("%d-%d-%d", a, b, r);
```

A 3-7-10

B 3-7-12

C 4-8-10

D 4-8-12

Qu'affiche le code suivant ?

```
int main() {
    float y = 'a';
    printf("%f", y);
    return 0;
}
```

- **A** a
- **B** Erreur de segmentation
- **C** a.00000
- **D** 97.00000

#QDLE#Q#ABCD*#50#

Qu'affiche le code suivant ?

```
int main() {
    float y = 'a';
    printf("%f", y);
    return 0;
}
```

A. a

B. Erreur de segmentation

C. a.00000

D. 97.00000

Qu'affiche le code suivant ?

```
int main() {
  int a = 1;
  int c = 3;
  int b = a << c;
  printf("%d\n", b);
  return 0;
}
```

A 8

B 1

C -724794438

D Erreur de segmentation

#QDLE#Q#A*BCD#35#

Qu'affiche le code suivant ?

```
int main() {
  int a = 1;
  int c = 3;
  int b = a << c;
  printf("%d\n", b);
  return 0;
}
```

- **A** 8
- **B** 1
- **C** -724794438
- **D** Erreur de segmentation

Quelle est la fonction qui permet de connaître la taille d'un tableau générique d'entiers ?

A strlen

B length

C sizeof

D Aucunes de ces réponses

#QDLE#Q#ABCD*#30#

Quelle est la fonction qui permet de connaître la taille d'un tableau générique d'entiers ?

A strlen

B length

C sizeof

D Aucunes de ces réponses

Prenons un tableau d'entiers *tab* de taille *l* non nulle. Que renvoie cette fonction ?

```
int f(int tab[], int l, int val){
  for (int i=0; i<l; i++) {
     if (tab[i]==val)
        printf("Trouvé !\n");
        return i;
  }
  return -1;
}
```

A Elle renvoie l'indice de la première occurrence de val dans tab, sinon -1

B Elle renvoie toujours 0

C Elle renvoie toujours -1

D Erreur de segmentation

#QDLE#Q#AB*CD#70#

Prenons un tableau d'entiers *tab* de taille *l* non nulle. Que renvoie cette fonction ?

```
int f(int tab[], int l, int val){
  for (int i=0; i<l; i++) {
     if (tab[i]==val)
        printf("Trouvé !\n");
        return i;
  }
  return -1;
}
```

A Elle renvoie l'indice de la première occurrence de val dans tab, sinon -1

B Elle renvoie toujours 0

C Elle renvoie toujours -1

D Erreur de segmentation

Chaînes de caractères

Laquelle de ces déclarations est incorrecte ?

A char[] str = “Hello world!”;

B char *str = “Hello world!”;

C char str[25] = “Hello world!”;

D char str[] = “Hello world!”;

#QDLE#Q#A*BCD#50#

Laquelle de ces déclarations est incorrecte ?

A char[] str = “Hello world!”;

B char *str = “Hello world!”;

C char str[25] = “Hello world!”;

D char str[] = “Hello world!”;

Quelle est la fonction qui permet de connaître la taille d'une chaîne de caractères ?

A sizeof

B length

C strlen

D Aucunes de ces réponses

#QDLE#Q#ABC*D#25#

Quelle est la fonction qui permet de connaître la taille d'une chaîne de caractères ?

A sizeof

B length

C strlen

D Aucunes de ces réponses

Qu'affiche le code suivant ?

```
int test(int c) {
  printf("c\n");
  return c;
}

int main() {
  test(2+3);
  return 0;
}
```

A. 5

B. c

C. Erreur de segmentation

D. Erreur à la compilation

#QDLE#Q#AB*CD#45#

Qu'affiche le code suivant ?

```
int test(int c) {
  printf("c\n");
  return c;
}

int main() {
  test(2+3);
  return 0;
}
```

A 5

B c

C Erreur de segmentation

D Erreur à la compilation

Quelle est la valeur de la variable *len* ?

```
int len = strlen("toto");
```

A 4 - Le zéro terminal de la chaîne n'est pas pris en compte.

B 5 - Le zéro terminal de la chaîne est pris en compte.

C 0 - La chaîne n'est pas correcte.

#QDLE#Q#A*BC#45#

Quelle est la valeur de la variable *len* ?

```
int len = strlen("toto");
```

A 4 - Le zéro terminal de la chaîne n'est pas pris en compte.

B 5 - Le zéro terminal de la chaîne est pris en compte.

C 0 - La chaîne n'est pas correcte.

Qu'affiche le code suivant ?

```
int main() {
  char s[] = {'t','o','\0','t','o','\0'};
  printf("%s\n", s);
  return 0;
}
```

A toto

B to\0to

C to

D Erreur à la compilation

#QDLE#Q#ABC*D#45#

Qu'affiche le code suivant ?

```
int main() {
  char s[] = {'t','o','\0','t','o','\0'};
  printf("%s\n", s);
  return 0;
}
```

A toto

B to\0to

C to

D Erreur à la compilation

Qu'affiche le code suivant ?

```
int main () {
    char* toto = "toto";
    toto[2] = 97;
    printf("%s\n", toto);
    return 0;
}
```

A. toto

B. to97to

C. toao

D. Erreur de segmentation

#QDLE#Q#ABCD*#45#

Qu'affiche le code suivant ?

```
int main () {
    char* toto = "toto";
    toto[2] = 97;
    printf("%s\n", toto);
    return 0;
}
```

A toto

B to97to

C toao

D Erreur de segmentation

Que fait cette fonction pour une chaîne de caractères a non vide ?

```
int func(char* a) {
  char* b = a;
  while(*++a)
    {};
  return a-b;
}
```

A. Une erreur de segmentation

B. Une différence entre caractères

C. Elle calcule la taille de la chaîne de caractères a

D. Elle renvoie un pointeur sur le début de la chaîne a

#QDLE#Q#ABC*D#60#

Que fait cette fonction pour une chaîne de caractères a non vide ?

```
int func(char* a) {
  char* b = a;
  while(*++a)
    {};
  return a-b;
}
```

- **A** Une erreur de segmentation
- **B** Une différence entre caractères
- **C** Elle calcule la taille de la chaîne de caractères a
- **D** Elle renvoie un pointeur sur le début de la chaîne a

À quoi correspond la fonction strcat proposée par <string.h> ?

A Elle permet d'imprimer un emoji en forme de chat dans la console

B Elle permet de concaténer le contenu d'une chaîne de caractères à une autre

C Elle découpe une chaîne en sous-chaînes selon un caractère de séparation

D Elle n'existe pas en réalité

#QDLE#Q#AB*CD#40#

À quoi correspond la fonction strcat proposée par <string.h> ?

A Elle permet d'imprimer un emoji en forme de chat dans la console

B Elle permet de concaténer le contenu d'une chaîne de caractères à une autre

C Elle découpe une chaîne en sous-chaînes selon un caractère de séparation

D Elle n'existe pas en réalité

Qu'affiche le code suivant ?

```
int main(){
   int a = 1;
   if ((char)a == '1')
     printf("%d\n", a);
   else
     printf("FAIL\n");
   return 0;
}
```

A. Erreur à la compilation

B. 1

C. La valeur ASCII de 1

D. FAIL

#QDLE#Q#ABCD*#60#

Qu'affiche le code suivant ?

```
int main(){
   int a = 1;
   if ((char)a == '1')
     printf("%d\n", a);
   else
     printf("FAIL\n");
   return 0;
}
```

A Erreur à la compilation

B 1

C La valeur ASCII de 1

D FAIL

Pointeurs

À quoi correspond la ligne de code suivante ?

```
int* a;
```

A Elle multiplie les contenus des variables int et a

B Elle déclare une variable a de type pointeur sur entier.

C Elle déclare une variable (opérateur *) a de type int.

D Elle déclare un pointeur générique sur 4 octets

#QDLE#Q#AB*CD#40#

À quoi correspond la ligne de code suivante ?

```
int* a;
```

A Elle multiplie les contenus des variables int et a

B Elle déclare une variable a de type pointeur sur entier.

C Elle déclare une variable (opérateur *) a de type int.

D Elle déclare un pointeur générique sur 4 octets

Qu'affiche le code suivant ?

```
char* result = "-" * 5;
printf("%s\n", result);
```

A Le code ASCII du caractère '-' est 45. Le résultat est donc 225.

B Erreur à la compilation

C -----

#QDLE#Q#AB*C#40#

Qu'affiche le code suivant ?

```
char* result = "-" * 5;
printf("%s\n", result);
```

A Le code ASCII du caractère '-' est 45. Le résultat est donc 225.

B Erreur à la compilation

C -----

Qu'affiche ce code sur un système classique 64bits ?

```
int main() {
  int x = 20000;
  double y = 26;
  int* p = &x;
  double* q = &y;
  printf("T(p)=%d\n", sizeof(p));
  printf("T(q)=%d\n", sizeof(q));
  return 0;
}
```

A
T(p) = 4
T(q) = 4

B
T(p) = 4
T(q) = 8

C
T(p) = 8
T(q) = 8

D
Erreur de segmentation

#QDLE#Q#ABC*D#70#

Qu'affiche ce code sur un système classique 64bits ?

```
int main() {
  int x = 20000;
  double y = 26;
  int* p = &x;
  double* q = &y;
  printf("T(p)=%d\n", sizeof(p));
  printf("T(q)=%d\n", sizeof(q));
  return 0;
}
```

A
T(p) = 4
T(q) = 4

B
T(p) = 4
T(q) = 8

C
T(p) = 8
T(q) = 8

D
Erreur de segmentation

Qu'affiche le code suivant ?

```
int main() {
  char* str = "y";
  char tab[1];
  tab[0] = 'y';
  printf("%d ",strlen(str));
  printf("%d" ,strlen(tab));
  return 0;
}
```

A 1 1

B 1 2

C 2 2

D 1 valeur indéfinie

#QDLE#Q#ABCD*#70#

Qu'affiche le code suivant ?

```
int main() {
  char* str = "y";
  char tab[1];
  tab[0] = 'y';
  printf("%d ",strlen(str));
  printf("%d" ,strlen(tab));
  return 0;
}
```

A 1 1

B 1 2

C 2 2

D 1 valeur indéfinie

Que risque très certainement d'afficher le code suivant ?

```
int main(){
    int a;
    int* ptr_a;
    *ptr_a = 1;
    printf("%d\n", a);
}
```

A 1

B valeur indéfinie

C Erreur à la compilation

D Erreur de segmentation

#QDLE#Q#ABCD*#45#

Que risque très certainement d'afficher le code suivant ?

```
int main(){
    int a;
    int* ptr_a;
    *ptr_a = 1;
    printf("%d\n", a);
}
```

A. 1

B. valeur indéfinie

C. Erreur à la compilation

D. Erreur de segmentation