



Concours d'accès en 1^{ère} année du cycle ingénieur
de la FST pour l'année universitaire 2019-2020
Epreuve d'Informatique
Durée 1h

Il faut répondre sur la grille. Chaque question a 2 lignes. En cas d'erreur sur la 1^{ère} ligne,
répondre sur la 2^{ème} ligne (ne pas utiliser blanco)

Sauf indication contraire, Une seule réponse est correcte pour chaque question
Aucun document autorisé. Calculatrice interdite

Question 1 : Un algorithme permet :

- (a) De mieux comprendre le fonctionnement des logiciels
- (b) Par une suite finie d'instructions de résoudre un problème.
- (c) De définir l'architecture d'un ordinateur
- (d) De communiquer avec des périphériques

Question 2 : Quel est l'intérêt de créer une variable? (2 réponses attendues)

- (a) Réduire le risque d'erreurs.
- (b) Garder des informations en mémoire.
- (c) Permettre de rendre les actions aléatoires.
- (d) Une fois créée elle peut être modifiée à tout moment.

Question 3 : Dans quel ordre s'effectue la compilation ?

- (a) Préprocesseur - Compilateur - Linker
- (b) Compilateur - Linker - Préprocesseur
- (c) Linker - Préprocesseur - Compilateur
- (d) Préprocesseur - Linker - Compilateur

Question 4 : Lequel des énoncés suivants n'est pas une déclaration de nom de variable valide?

- (a) int _a7;
- (b) int a_7;
- (c) int 7_a;
- (d) int _7a

Question 5 : Parmi les affirmations suivantes, laquelle est juste ?

- (a) une fonction ne peut pas être déclarée sans paramètres.
- (b) une fonction peut avoir des variables locales.
- (c) une fonction paramétrée ne peut pas appeler une autre fonction.
- (d) une fonction doit avoir une valeur de retour

Question 6 : De quel(s) type(s) sont les paramètres passés à une fonction ?

- (a) de type simple ou structuré
- (b) de type structuré
- (c) de type simple
- (d) cela dépend de la fonction



Question 7 : Quelle affirmation concernant les fonctions est juste ?

- (a) il faut toujours passer au moins un paramètre
- (b) il faut toujours affecter une valeur à l'identificateur de la fonction
- ☒ (c) une fonction est toujours appelée dans une instruction
- (d) une fonction peut utiliser une variable déclarée dans une autre fonction

Question 8 : Que dire de vrai à propos des variables globales ?

- (a) elles sont connues uniquement dans le sous-programme.
- ☒ (b) elles peuvent être utilisées par le programme principal et tous ses sous-programmes.
- (c) elles sont connues uniquement dans le programme principal.
- (d) elles peuvent être remplacées par des variables locales

Question 9 : Que dire de faux à propos des variables locales ?

- ☒ (a) elles sont connues par le programme principal.
- (b) elles sont connues uniquement dans le sous-programme où elles sont déclarées.
- (c) elles cachent les variables globales du même nom.
- (d) elles peuvent être de même nom dans différentes fonctions

Question 10 : Si on utilise une boucle POUR, dans un sous-programme, alors le compteur :

- (a) doit être déclaré dans la boucle.
- ☒ (b) doit être déclaré avant la boucle.
- (c) doit être incrémenté par 1.
- (d) peut être décrémenté par 1.

Question 11 : Lequel des éléments suivants n'est pas une déclaration de pointeur ?

- (a) `char *str;`
- (b) `char a[10];`
- ☒ (c) `char a;`
- (d) `char a[] = {'10', '20', '30', '40'};`

Question 12 : la variable A contient 10.5, Qu'est-ce que la ligne suivante affiche ?

- `printf("A = %d", A);`
- (a) `A = %d`
 - (b) `A = 10.5`
 - (c) `A = 10`
 - ☒ (d) Erreur

Question 13 : Combien de fois passe-t-on dans la boucle suivante ?

- ```
int Cpt = 10;
while ((Cpt < 11) { printf("bonjour\n"); }
```
- (a) 0
  - ☒ (b) 1
  - (c) 10
  - (d) c'est une boucle infinie

Question 14 : Combien vaudra Res après l'opération  $Res = (13/5) - 2$  ?

- (a) -2
- (b) 0
- ☒ (c) 0,6
- (d) 2

Question 15 : Combien vaudra Res après les opérations suivantes ?

- ```
int A = 4;
Res = 5 + A++;
Res += 2 + A;
Res -= 4 + (-A)
Res = Res + A++;
```
- (a) 9
 - (b) 10
 - (c) 12
 - (d) 14



Question 16 : Quelle est la sortie de ce code ?

```
#include <stdio.h>

void main()
{
    int var = 6;
    float var = 6;
    printf("%d", var)
}
```

- (a) 6.0000000 (b) 6.8 (c) 6 (d) Erreur de compilation

Question 17 : Une variable déclarée dans une fonction peut être utilisée dans la fonction principale main() ?

- (a) Faux (b) Vrai
(c) Vrai s'il est déclaré statique (d) Aucun de ces réponses

Question 18 : Quelle est la bonne méthode d'initialisation d'un tableau d'entiers ?

- (a) int tableau [4] = 10,5,8,9; (b) int tableau [4] = [10,5,8,9] ;
(c) int tableau [4] = {10,5,8,9}; (d) int tableau [4] = (10,5,8,9);

Question 19 : Par quelle valeur doit-on initialiser un pointeur ?

- (a) 0 (b) NULL (c) -1 (d) void ou rien

Question 20 : Que donne le code suivant : &A ?

- (a) L'adresse de la variable A (b) La valeur de la variable A
(c) La valeur de la variable sur laquelle pointe A (d) Null

Question 21 : Que donne le code suivant : *A ?

- (a) L'adresse de la variable A (b) La valeur de la variable A
(c) La valeur de la variable sur laquelle pointe A (d) Null

Question 22 : Quelle est la valeur de « z » dans ce code ?

```
#include <stdio.h>

int main()
{
    int x = 10, y = 5, w = 5;
    int d;
    int z = x == (y + w);
    printf("Valeur de z est %d", z);
}
```

- (a) 5 (b) 1 (c) 10 (d) Erreur de syntaxe

Question 23 : Soit le code suivant :

```
#include <stdio.h>
#define TAB_LENGTH 3

int main() {
    int tab[TAB_LENGTH];
    int j = 0;
    int *ptr = tab;
    for(, j < TAB_LENGTH, j++)
        tab[j] = 5;
    *(ptr + 1) = 3;
    printf("[ %d %d %d ]\n", tab[0], tab[1], tab[2]);
    return 0;
}
```

Le programme :

- (a) provoque une erreur (b) affiche [5 5 5]
(c) affiche [3 5 5] (d) affiche [5 3 5]

Question 24 : Soit le code suivant :

```
#include <stdio.h>

int main() {
    int a, b;
    int *ptr1, *ptr2;
    a = 5;
    b = a;
    ptr1 = &a;
    ptr2 = ptr1;
    b = (*ptr2)++;
    printf("a = %d, b = %d, *ptr1 = %d, *ptr2 = %d\n", a, b, *ptr1, *ptr2);
    return 0;
}
```

Le programme :

- (a) provoque une erreur (b) affiche a = 6, b = 5, *ptr1 = 6, *ptr2 = 6
(c) affiche a = 5, b = 5, *ptr1 = 5, *ptr2 = 5 (d) affiche a = 6, b = 6, *ptr1 = 6, *ptr2 = 6

Question 25 : Si on définit une variable comme ceci :

```
struct paire {
    int premier ;
    int second ;
};
struct paire v ;
```

alors on peut accéder au premier élément de v comme ceci :

- (a) v->premier (b) v.premier (c) v[0] (d) premier(v)