



Développement : Les fonctions

© tv <tvaira@free.fr> v.1.0

Ce sujet comprend 20 questions pour un total de 20 points.

A Questions de cours

Question 1 (1 point)

Quels sont les avantages de l'approche fonctionnelle (ou de la programmation procédurale) ?

Question 2 (1 point)

Quelles sont les différences entre les concepts de fonctions et de procédures ?

Question 3 (1 point)

À quoi sert une déclaration de fonction ? À quelle phase du processus de fabrication est-elle utile ?

Question 4 (1 point)

À quoi sert une définition de fonction ? À quelle phase du processus de fabrication est-elle utile ?

Question 5 (1 point)

Quelle est la différence entre un prototype et une signature ?

Question 6 (1 point)

Dans quels fichiers place-t-on les déclarations de fonction ?

Question 7 (1 point)

Dans quels fichiers place-t-on les définitions de fonction ?

Question 8 (1 point)

Quel est le coût d'un passage de paramètres par valeur ?

Question 9 (1 point)

Quelle(s) technique(s) peut-on utiliser pour optimiser le passage de paramètres par valeur ?

Question 10 (1 point)

Que permet la surcharge des fonctions en C++ ?

Question 11 (1 point)

Que se passe-t-il après un `return` dans une fonction ?

- Le programme s'arrête
- La fonction s'arrête et renvoie le résultat indiqué dans le `return`
- La fonction continue et ne renvoie pas de résultat

Question 12 (1 point)

Dans quel cas l'instruction `return` n'est pas obligatoire ?

- Quand la fonction ne prend aucun paramètre en entrée
- Quand la fonction est de type `void`
- Quand la fonction doit renvoyer la valeur 0

Question 13 (1 point)

Que représentent les paramètres d'une fonction ?

- Des variables qu'on lui envoie pour qu'elle puisse travailler dessus
- Des indications complémentaires sur le rôle de la fonction
- Des indications sur la valeur qu'elle doit renvoyer

Question 14 (1 point)

Laquelle de ces affirmations est fausse ?

- Une fonction n'est pas obligée de renvoyer une valeur
- Une fonction peut renvoyer une valeur de n'importe quel type
- Une fonction peut renvoyer plusieurs type de valeurs

Soit le code source suivant :

```
int main (void)
{
    long r, x = 2;
    r = carre(x);
    return 0;
}
```

Question 15 (1 point)

Quel sera le prototype de la fonction `carre()` ?

- `void carre(long);`
- `long carre(long);`
- `double carre(double);`
- `long carre(long x * long x);`
- `long carre(int);`
- Aucune des réponses proposées

Lorsqu'on saisit des arguments après le nom d'un programme, ceux-ci sont passés en paramètres de la fonction principal `main()`. Le prototype de la fonction `main()` d'un programme C/C++ est la suivante :

```
int main(int argc, char *argv[]); // ou : int main(int argc, char **argv);
```

- Le paramètre `argc` contient le nombre d'arguments (`%d` pour l'afficher).

- Le paramètre `argv` contient l'ensemble des arguments sous la forme de chaînes de caractères : `argv[0]` contient le nom du programme, `argv[1]` contient le premier argument, etc ... (`%s` pour afficher un `argv[x]`).

Soit le code source du programme `testArg` :

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int i;

    printf("nb d'arguments = %d\n", argc);
}
```

```
//printf("nom du programme argv[0] : %s\n", argv[0]);

for(i=0;i<argc;i++)
{
    printf("argv[%d] = %s\n", i, argv[i]);
}

return 0;
}
```

Question 16 (1 point)

Quel sera l'affichage obtenu lorsqu'on exécute la commande : `./testArg_bts_2016` ?

- nb d'arguments = 1
argv[0] = ./testArg_bts_2016
- nb d'arguments = 2
argv[0] = ./testArg
argv[1] = bts_2016
- nb d'arguments = 2
argv[0] = bts
argv[1] = 2016
- nb d'arguments = 3
argv[0] = ./testArg
argv[1] = bts
argv[2] = 2016

Question 17 (1 point)

Que réalise l'appel d'une fonction ?

Question 18 (1 point)

Donner le prototype de la fonction `recherche()` qui admet 3 paramètres : un tableau d'entiers, le nombre d'éléments contenus dans le tableau, une valeur à rechercher dans le tableau et qui détermine si la valeur à rechercher est présente ou non. La fonction retournera la position (entre 0 et n) de la valeur dans le tableau si elle existe et -1 sinon.

Question 19 (1 point)

Écrire l'appel de la fonction `recherche()` lorsque le programme ci-dessous affiche "position = 5". Soit le code source du programme `testArg` :

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int t[] = {1, 3, 4, 6, 9, 11, 12, 13, 15, 27, 29, 30, 31, 41, 42, 49};
    int position;

    // Appel pour rechercher une valeur

    printf("position = %d\n", position);

    return 0;
}
```

Question 20 (1 point)

Dans le programme ci-dessus, que risque-t-il de se passer si l'on indique un nombre d'éléments erronés contenus dans le tableau ?