

# DS: Programmation C

## 30 Novembre 2023 - 08h30/10h00

Nom: ..... Prénom: .....

Documents autorisés : une feuille A4 recto/verso manuscrite.

### Structures de base du langage

**Question 1** Considérez le programme suivant :

```
#include <stdio.h>

int main()
{
    int i=2;
    int j=++i +i;
    printf("%d\n",j); return 0;
}
```

Parmi les propositions suivantes laquelle correspond à ce que le programme va afficher ?

- 6
- 5
- 4
- erreur de compilation

**Question 2** Considérez le programme suivant :

```
#include <stdio.h>

int main()
{
    int i=10, j=0;
    if (i || (j=l + 10))
        // programme dans lequel j n'apparait pas
        ; return 0;
}
```

- 0
- 20
- Dépend de l'implantation du langage.
- Erreur de compilation.

**Question 3** Pour quelle déclaration, l'expression "PI++" produira une erreur?

- float PI=3.14
- char \*PI="A"
- #define PI 3.14
- Aucune des lignes ci-dessus.

**Question 4** Considérez le programme suivant :

```
#include <stdio.h>

int main()
{
    int i=065, j=65;
    printf("%d %d",i,j);
}
```

- 53 65
- 65 65
- 065 65
- 053 65
- erreur de syntaxe.

## Pointeurs et tableaux

**Question 5** Considérez le programme suivant :

```
#include <stdio.h>

void main() {
    int a[] = {1,2,3,4,5}, *p;
    p = a;
    ++*p;
    printf("%d", *p);
    p += 2;
    printf("%d", *p);
}
```

Qu'est-ce qui sera imprimé à l'écran?

- 24
- 34
- 22
- 23

**Question 6** Considérez le programme suivant :

```
#include <stdio.h>

void main() {
    char *p;
    p = "Hello";
    printf("%c\n", *&*p);
}
```

Qu'est-ce qui sera imprimé à l'écran?

- Hello
- H
- Une adresse va être imprimée
- Aucun des choix précédents

**Question 7** Considérez le programme suivant :

```
#include <stdio.h>
int main(){
    int ary[4] = {1, 2, 3, 4};
    int *p = ary + 3;
    printf("%d\n", p[-2]);
}
```

Qu'est-ce qui sera imprimé à l'écran?

- 1
- 2
- Une valeur non spécifiée
- Une erreur de compilation

**Question 8** Considérez le programme suivant :

```
#include <stdio.h>
void main(){
    char s[] = "hello";
    s++;
    printf("%c\n", *s);
}
```

Qu'est-ce qui sera imprimé à l'écran?

- h
- e
- o
- Une erreur de compilation

## Fonctions

**Question 9** Considérez le programme suivant :

```
#include <stdio.h>
void reverse(int i);
int main(){
    reverse(1);
}
void reverse(int i){
    if (i > 5)
        return ;
    printf("%d ", i);
    return reverse(++i);
}
```

Que va produire ce programme ?

- 1 2 3 4 5
- Segmentation fault
- 5 4 3 2 1
- Une erreur de compilation

**Question 10** Considérez le programme suivant – similaire au précédent mais on a changé le ++i en i++ dans la dernière ligne :

```
#include <stdio.h>
void reverse(int i);
int main(){
    reverse(1);
}
void reverse(int i){
    if (i > 5)
        return ;
    printf("%d ", i);
    return reverse(i++);
}
```

Que va produire ce programme ?

- 1 2 3 4 5
- Segmentation fault
- 5 4 3 2 1
- Une erreur de compilation

**Question 11** Considérez le programme suivant :

```
#include <stdio.h>
int *init(int i);
int main(){
    int *t;
    int i;
    init(5);
    for (i=0;i<5;i++) printf("%d ",t[i]);
}
int *init(int i){
    int tab[i];

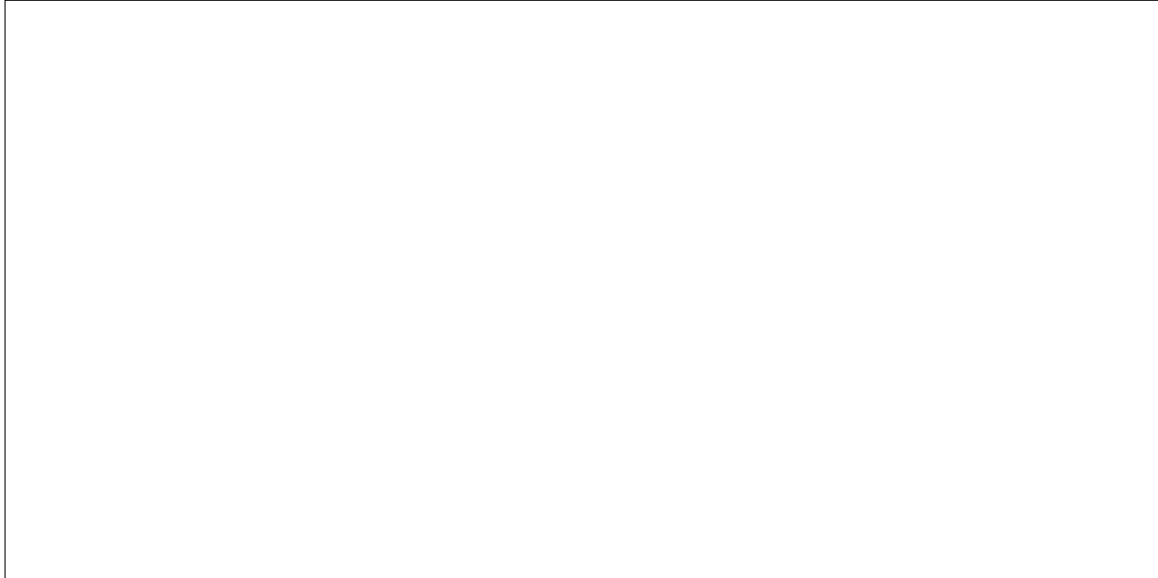
    for (i=0;i<5;i++) tab[i]=i+1;
    return tab;
}
```

- 1 2 3 4 5
- Segmentation fault
- imprime 4 valeurs indéfinies
- Une erreur de compilation

**Question 12** Ecrivez un programme qui imprime l'argument suivant "-o" dans la liste des arguments qui sont donnés. Par exemple si le programme compilé est appelé `impression_argument` alors la commande

`./impression_argument il y a des arguments qui -o coucou` comptent pas les autres  
tappée dans un terminal produira

`coucou`



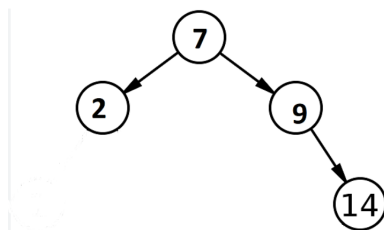
## Problème : arbres binaires

On considère des arbres binaires contenant des entiers comme structure de données. Pour représenter un arbre on donnera un pointeur vers une structure qui contient la donnée (un entier) et deux pointeurs (un vers le fils gauche et un autre vers le fils droit). L'arbre vide sera représenté comme le pointeur NULL.

L'implantation du type `b_tree` est la suivante

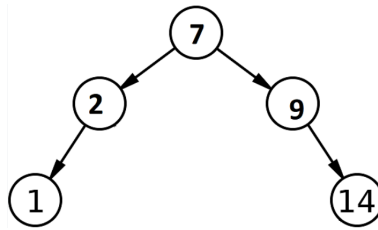
```
struct b_tree{
    int data;
    struct b_tree *fg,*fd;
};
typedef struct b_tree b_tree;
```

**Question 13** Donner dans le cadre les lignes de codes C qui permettent de représenter l'arbre A suivant :



**Question 14** Donner dans le cadre qui suit les lignes de codes C qui permettent d'implanter une fonction `int nombre(b_tree *a)` qui compte le nombre de noeud de l'arbre passé en paramètre :

**Question 15** Donner dans le cadre qui suit les lignes de codes C qui permettent d'implanter une fonction qui prend en paramètre un arbre, un entier et qui ajoute le paramètre dans l'arbre le plus en bas à gauche possible. Par exemple si on ajoute 1 à l'arbre A donné en exemple précédemment à la question 13 on doit obtenir l'arbre :



**Question 16** Utilisez le cadre suivant pour insérer toutes les indications utiles que vous n'avez pas eu la place de mettre avant. Notez bien le numéro de la question suivi du commentaire que vous jugez bon d'ajouter.

