



+52/1/18+

UNIVERSITÉ CÔTE D'AZUR

POLYTECH

2021-2022

Nom :

LATAPIE

Prénom:

Florian

Programmation Procédurale – SI3

Mardi 14 décembre 2021

Question 1: Si on a `char *t[] = {"Hello", "world"};` quelle est la valeur de `*(t+1)`?

☐ 'l'

☒ "world"

☐ 'x'

☐ "ello"

Question 2: Si on appelle le programme `prog` de la façon suivante: `$./prog "Hello world"` Quelle est la valeur de `argv[2]`?

☒ NULL

☐ Hello world

☐ world

☐ Hello

Question 3: Si on appelle le programme `prog` de la façon suivante: `$./prog foo bar gee hux` Quelle est la valeur de `argc` et `argv[3]`?

☐ 4, hux

☒ 5, gee

☐ 4, gee

☐ 5, hux

Question 4: En C, la notation `t[4]` est équivalente à:

☐ `t+4`

☒ `*(t)+4`

☒ `*(t+4)`

☐ `*t+4`

Question 5: Si on a `int t[] = {4,6,7,8};` que vaut `t[t]`?

☐ 6

☒ c'est une erreur

☒ 4

Question 6: Soit la déclaration `int *t;` alors l'expression `t[*t]` est:

☐ correcte et son type est `int *`

☐ correcte et son type est `int **`

☒ correcte et son type est `int`

☐ syntaxiquement incorrecte

Question 7: Soit les déclarations suivantes: `char p1[]="abcd", p2[]="WXYZ";`

☒ On peut écrire `p1[0]='A';`

☒ On peut écrire `*p1='A';`

☒ On peut écrire `p2=p1;`

☒ `sizeof(p1)` vaut 5

Question 8: Si `t` est défini par `int *t`, quel est alors le type de `t+2`?

☒ `(int *)`

☐ `(int **)`

☐ c'est une erreur

☐ `(int)`

Question 9: Si on a `char *t[] = {"Hello", "world"};` quelle est la valeur de `(**t)+1`?

☐ "world"

☐ 'x'

☐ "ello"

☒ 'l'

Question 10: La chaîne de caractères "une chaîne" est de type

☐ `char`

☒ `char *`

☐ `string`

☒ `const char *`

Question 11: Soit la déclaration `char str[]="hello";` Que vaut `*str`?

☒ 'h'

☐ NULL

☐ "hello"

☐ Erreur



Question 12: Si on déclare `int *v2`; que vaut alors `*v2`:

- 1/1 ☐ l'adresse de v2 ☒ l'entier pointé par v2 ☐ la valeur de v2

Question 13: Soit les déclarations suivantes: `char *p1="abcd"`, `*p2 = "WXYZ"`;

- 0.3333/1 ☒ On peut écrire `*p1='A'`; ☐ `sizeof(p1)` vaut 5
☒ On peut écrire `p2=p1`; ☒ On peut écrire `p1[0]='A'`;

Question 14: Si on déclare `int v1`; que vaut alors `&v1`:

- 1/1 ☒ l'adresse de v1 ☐ l'entier pointé par v1 ☐ la valeur de v1

Question 15: Soient les déclarations suivantes: `int n=100`, `*p=&n`;
Si `n` est stocké à l'adresse 3000 et `p` à l'adresse 3004, quelle sera la valeur affichée par `printf("%d",*p)`;

- 1/1 ☐ 3004 ☐ 3000 ☐ indéterminée ☒ 100

Question 16: Soient les déclarations suivantes `int n=100`, `*p=&n`, `**q=&p`; Que vaut `q` ?

- 1/1 ☐ la valeur de p ☒ l'adresse de p ☐ l'adresse de n ☐ la valeur de n

Question 17: Soit le code suivant:

```
#define PI int *  
PI x, y;
```

- 0.2/1 ☐ il y a une erreur à l'exécution ☒ x et y sont de type pointeur
☐ il y a une erreur de compilation ☒ x est de type pointeur et y est un entier
☐ x et y sont de type entier ☐ x est de type entier et y est un pointeur

Question 18: Soit le code suivant:

```
typedef int * PI;  
PI x, y;
```

- 1/1 ☐ il y a une erreur de compilation ☒ x et y sont de type pointeur
☐ x est de type pointeur et y est un entier ☐ x est de type entier et y est un pointeur
☐ x et y sont de type entier ☐ il y a une erreur à l'exécution

Question 19: Si `t` est défini par `int *t`, quel est alors le type de `&t`

- 0.3333/1 ☐ `(int *)` ☒ `(int)` ☐ c'est une erreur ☒ `(int **)`

Question 20: Si on a `char *t[] = {"Hello", "world"};` quelle est la valeur de `(*t)+1`?

- 0.3333/1 ☒ "world" ☐ 'x' ☐ 'l' ☒ "ello"