

## Examen – Session principale

**Matière :** Atelier programmation I

**Enseignants :** Majdi Jribi et Imen Ben Ida

**Filière :** MPI

**Nombre de pages :** 08 pages

**Semestre:** premier semestre

**Date:** 08 Janvier 2019

**Durée:** 1h30

**Documents :** non autorisés

Les réponses doivent être rédigées obligatoirement sur les feuilles de réponse (pages 7 et 8)  
 L'examen contient 8 pages. Seulement les pages 7 et 8 sont à rendre.

### Répondre sur les feuilles de réponse aux questions suivantes

- 1- Soit la déclaration suivante :

```
char les[] = "ab" ;
```

Que vaut `les[2]` ?

- 2- Donner la déclaration d'un tableau dont chacun de ses 20 éléments est une structure à trois champs :

- Nombre : un entier
- Ptring : un pointeur sur entier
- Tab : un tableau de 15 caractères.

Supposant que ce tableau est initialisé, donnez l'expression permettant de placer dans le champ Ptring du 4ème élément du tableau, l'adresse du champ Nombre du 11ème élément.

- 3- Dans le code suivant :

```
char ch[] = "Bonjour", *pc = ch ;
while (*pc) pc++;
```

Que vaut `pc` après l'exécution de l'instruction `while` ? Vers quel caractère pointe-t-il ?

- 4- Soit le code suivant :

```
#include<stdio.h>
void g(int *p) {
  *p = 10 ;
}
void main() {
  int *p ;
  g(p) ;
  printf("%d\n", *p) ;
}
```

Qu'est ce qui s'affiche ?

5- Dans le code suivant :

```
int i=0, j=0 ;
```

```
j = i++ ;
```

Que vaut la variable j après l'instruction d'affectation ?

6- Soit le code suivant :

```
char tab[] = "bonjour" ;
```

```
int i ;
```

```
for (i=0 ; tab[i] != '\0' ; i++) {
```

```
tab[i] = tab[i] - 'a' + 'A' ; }
```

Que fait ce programme ?

Proposez une version sans utiliser l'opérateur d'accès tableau ([])

7- Soit le code suivant :

```
int i = 0, j = 5, somme = 20 ;
```

```
if (j && (i = somme / j) ) somme = 10 ;
```

```
else somme = 30 ;
```

Donnez les valeurs des variables après l'exécution de ce programme.

8- Donner deux déclarations différentes d'un pointeur sur un tableau de 10 chaînes de caractères.

9- Soit le code suivant :

```
#include <stdio.h>
```

```
void main() {
```

```
int i = -1, j = -i;
```

```
int w1, w2;
```

```
w1 = (i > 0) && (j < 0) || (i < 0) && (j > 0);
```

```
w2 = (i <= 0) || (j >= 0) && (i >= 0) || (j <= 0);
```

```
printf("%d", w1 == w2);
```

```
}
```

Qu'est ce qui s'affiche ?

10- Soit le code suivant :

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void main() {
```

```
char t[20] = "ABCDEFGHIJK";
```

```
int s = strlen(t);
```

```
t[3] = '\0';
```

```
puts(t) ;
```

```
s += strlen(t);
```

```
strcpy(t, "ABCDEF");
```

```
s += strlen(t);
```

```
strcat(t, "ABC");
```

```
puts(t) ;
```

```
s += strlen(t);
```

```
printf("%d", s); }
```

Qu'est ce qui s'affiche ?

11- Soit le code suivant :

```
#include <stdio.h>
#include <stdlib.h>
void main() {
    char *p;
    int i;
    p = (char *)malloc(10*sizeof(char));
    for(i = 0; i < 10; i++)
        p[i] = 'A' + i;
    printf("%c", *(p+9));
    free(p);
}
```

Qu'est ce qui s'affiche ?

12- Soit le code suivant :

```
#include<stdio.h>
int add(int par) {
    par += par;
    return par;
}
int add2(int p1, int p2) {
    return p1 + p2;
}
void main()
{
    int var = 0;
    var = add2(add(2), add(4));
    var = add2(var, var);
    printf("%d", var);
}
```

Qu'est ce qui s'affiche ?

13- Soit le code suivant :

```
#include<stdio.h>
void main()
{
    int a;
    int b;
    char tab[]="123abc 456.65";
    sscanf(tab, "%d%s%d", &a, &b);
    printf("%d\n%d\n", a, b);
}
```

Qu'est ce qui s'affiche ?

14- Soit le code suivant :

```
#include<stdio.h>
void main()
{
    int a=1234;
    char ch1[]="abcd";
    char ch[15];
    sprintf(ch,"%2d%2s",a,ch1);
    puts(ch);
}
```

Qu'est ce qui s'affiche ?

15- Soit le code suivant :

```
#include<stdio.h>
void main()
{
    int a=1234;
    char ch1[]="abcd";
    char ch[15];
    sprintf(ch,"%d%2.3s",a,ch1);
    puts(ch);
}
```

Qu'est ce qui s'affiche ?

16- Soit le code suivant:

```
#include<stdio.h>
void main()
{
    int a=1234;
    char ch1[]="abcd";
    char ch[15];
    sprintf(ch,"%6d%6.8s",a,ch1);
    puts(ch);
}
```

Qu'est ce qui s'affiche ?

17- Soit le code suivant:

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    double a;
    char ch[]="123.5aze";
    a=atof(ch);
    printf("%f",a);
}
```

Qu'est ce qui s'affiche ?

18- Soit le code suivant:

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int a;
    char ch[]="123.5az";
    a=atoi(ch);
    printf("%d",a);
}
```

Qu'est ce qui s'affiche ?

19- Soit le code suivant:

```
#include <stdio.h>
void main()
{char CH[101];
  char *PCH;
  int ABC[26];
  int *PABC;
  printf("Entrez une ligne de texte(max.100 caractères)\n");
  gets(CH);
  for (PABC=ABC; PABC<ABC+26; PABC++)
      *PABC=0;
  for (PCH=CH; *PCH; PCH++){
      if (*PCH>='A' && *PCH<='Z')
          (*(ABC+(*PCH-'A')))+++;
      if (*PCH>='a' && *PCH<='z')
          (*(ABC+(*PCH-'a')))+++;
  }
  for (PABC=ABC; PABC<ABC+26; PABC++)
      if (*PABC)
          printf(" %d fois \t: '%c' \n", *PABC, 'A'+(PABC-ABC));
}
```

C'est quoi le rôle de ce programme ?

20- Soit le code suivant:

```
#include <stdio.h>
#include <string.h>
void main()
{
    char CH1[101], CH2[101];
    char *P1, *P2;
    int TROUVE;

    printf("Entrez la première chaîne de caractères:\n");
    gets(CH1);
    printf("Entrez la deuxième chaîne de caractères:\n");
    gets(CH2);
}
```



```

for (P2=CH2; *P2; P2++)
{
    TROUVE = 0;
    for (P1=CH1 ; *P1 && !TROUVE ; P1++)
        if (*P2 == *P1)
        {
            TROUVE = 1;
            strcpy(P1, P1+1);
        }
    }
    printf("Chaîne résultat : \"%s\" \n", CH1);
}

```

C'est quoi le rôle de ce programme ?

21- C'est quoi le rôle de programme suivant ?

```

#include<stdio.h>
void main()
{
    int i,j;
    int t1[10];
    int t2[10];
    for (i=0 , j=0 ; j<10 ;j++)
        if (t2[j] >0 ) t1[i++]= t2[j],
    for (j=i ; j<10 ;j++) t1[j]= 0 ;
}

```