

# **Correction d'examen session principale**

## **Atelier programmation I**



## 1-

Soit la déclaration suivante : `char les[] = "ab".`

Que vaut `les[2]` ?

Réponse : `les[2]` vaut `'\0'`.

Le tableau `les` est une chaîne de caractères, car il a été initialisé avec une chaîne (`"ab"`).

## 2-

(Donner la déclaration d'un tableau dont chacun de ses 20 éléments est une structure à trois champs : un entier nombre, un pointeur d'entier `puint` et un tableau de 15 caractères. Supposant ce tableau initialisé, donnez l'expression permettant de placer dans le champ `puint` du 4ème élément du tableau, l'adresse du champ nombre du 11ème élément.

Réponse :

```
tableau[3].puint = &tableau[10].nombre
```

Avec les déclarations suivantes :

```
typedef struct examen_t {  
    int nombre ;  
    int puint ;  
    char chaine [15] ;  
} Examen ;  
Examen tableau[20] ;
```

## 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 ?

Réponse : `pc` vaut `&pc[7]` et le caractère pointé est le caractère de `?\n` de chaîne `'\0'`.

L'instruction `while` peut s'écrire ainsi :

```
while ( *pc != '\0' ) {  
    pc = pc + 1 ;  
}
```

Comme `pc` est initialisé avec l'adresse du premier caractère de la chaîne `ch`, la boucle va être exécutée tant que le caractère pointé n'est pas le caractère de `?\n` de chaîne. À la sortie de la boucle, `pc` vaut `&pc[7]` et le caractère pointé est le caractère de `?\n` de chaîne `'\0'`.

## 4-

Soit le code suivant :

```
void g(int *p) {  
    *p = 10 ;  
}  
int main() {  
    int *p ;  
    g(p) ;  
    printf("%d\n", *p) ;  
}
```

Qu'est ce qui s'affiche ?

Réponse : L'affichage est indéterminé ou erreur d'exécution.

Le pointeur `p` défini dans la fonction `main` n'est pas initialisé. Comme il s'agit d'une variable locale, son contenu est indéterminé.

L'expression `g(p)`, puis, dans la fonction `g`, `pp = 10`, sont donc erronées. Avec de la chance, il y aura une erreur à l'exécution (segmentation error), ou pire, un écrasement mémoire.

## 5-

Soit le code suivant :

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

Que vaut la variable `j` après l'instruction d'effection ?

Réponse : `j` vaut 0 ..

## 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 ([]).

Réponse :

Ce programme transforme la chaîne de caractères "bonjour" en majuscules.

```
char tab[] = "bonjour",pc;
for (pc = tab;pc!='\0'; pc++) {
    *pc =pc - 'a' + 'A';
}
```

## 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.

Réponse : i vaut 4, j vaut 0 et somme vaut 10.

Le test dans l'instruction if peut s'écrire ainsi :

```
j != 0 && (i = somme / j) != 0
```

Comme j est différent de zéro (il vaut 5), le premier terme est vrai. Le résultat de l'expression somme/j est d'abord calculé, donc la valeur 4 est stockée dans i, qui différente de 0, entraîne que le second terme est vrai également.

En conséquence, i vaut 4, j vaut 0 et somme vaut 10.

## 8-

Donner la déclaration d'un pointeur sur un tableau de 10 chaînes de caractères.

Réponse :

Deux possibilités, selon l'utilisation des « chaînes de caractères » :

- (a) Un tableau de caractères : `char (*pointeur)[10][LONG];`
- (b) Un pointeur : `char*(*pointeur)[10];`

**9-**

**Soit le code suivant :**

```
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 ?

Réponse : affiche 1

**10-**

**Soit le code suivant :**

```
#include <stdio.h>
#include <string.h>
int main(void) {
    char t[20] = "ABCDEFGHGIJK";
    int s = strlen(t);
    t[3] = '\0';
    s += strlen(t);
    strcpy(t, "ABCDEF");
    s += strlen(t);
    strcat(t, "ABC");
    s += strlen(t);
    printf("%d", s);
    return 0;
}
```

Qu'est ce qui s'affiche ?

Réponse : affiche 29

**11-**

**Soit le code suivant :**

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

Qu'est ce qui s'affiche ?

Réponse : affiche J

## 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;
}
int main(void) {
    int var = 0;
    var = add2(add(2), add(4));
    var = add2(var, var);
    printf("%d", var);
    return 0;
}
```

Qu'est ce qui s'affiche ?

Réponse : affiche 24

## 13-

Soit le code suivant :

```
#include <stdio.h>

#include <string.h>

int main(void) {

    int a;

    int b;

    char tab[]="123abc 456.65";

    sscanf(tab, "%d%s%d", &a, &b);

    printf("%d\n%d\n", a, b);

    return 0;

}
```

Qu'est ce qui s'affiche ?

Réponse : affiche

123

456

## 14-

Soit le code suivant :

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

Qu'est ce qui s'affiche ?

Réponse : affiche "1234abcd"

## 15-

Soit le code suivant :

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

void main (void) {

    int a=1234;

    char ch1[]="abcd";

    char ch[15];

    sprintf(ch,"%d%2.3s",a,ch1);

    puts(ch);

}
```

Qu'est ce qui s'affiche ?

Réponse : affiche "1234abc"

## 16-

Soit le code suivant :

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

Qu'est ce qui s'affiche ?

Réponse : affiche " 1234 abcd"

## 17-

Soit le code suivant :

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

Qu'est ce qui s'affiche ?

Réponse : affiche 123.500000

## 18-

Soit le code suivant :

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

void main (void) {

    int a;

    char ch[]="123.5ax";

    a=atoi(ch);

    printf("%d",a);

}
```

Qu'est ce qui s'affiche ?

Réponse : affiche 123

## 19-

Soit le code suivant :

```
#include <stdio.h>

main()

{
```

```

/* Déclarations */
char CH[101]; /* chaîne donnée */
char *PCH; /* pointeur d'aide dans CH */
int ABC[26]; /* compteurs des différents caractères */
int *PABC; /* pointeur d'aide dans ABC */

/* Saisie des données */
printf("Entrez une ligne de texte (max.100 caractères) :\n");
gets(CH);

/* Initialiser le tableau ABC */
for (PABC=ABC; PABC<ABC+26; PABC++)
    *PABC=0;

/* Compter les lettres */
for (PCH=CH; *PCH; PCH++)
{
    if (*PCH>='A' && *PCH<='Z')
        (*(ABC+(*PCH-'A')))+++; /* Attention aux parenthèses! */
    if (*PCH>='a' && *PCH<='z')
        (*(ABC+(*PCH-'a')))+++;
}

/* Affichage des résultats */
/* (PABC-ABC) est le numéro de la lettre de l'alphabet. */
printf("La chaîne \"%s\" contient :\n", CH);
for (PABC=ABC; PABC<ABC+26; PABC++)
    if (*PABC)
        printf(" %d\tfois la lettre '%c' \n",
                *PABC, 'A'+(PABC-ABC));

return 0;
}

```

Réponse :

**Programme qui lit une chaîne de caractères CH au clavier et qui compte les occurrences des lettres de l'alphabet en ne distinguant pas les majuscules et les minuscules.**



```
Entrez une ligne de texte (max.100 caractères) :
correction d'examen session principale
La chaîne "correction d'examen session principale" contient :

2      fois la lettre 'A'
3      fois la lettre 'C'
1      fois la lettre 'D'
5      fois la lettre 'E'
4      fois la lettre 'I'
1      fois la lettre 'L'
1      fois la lettre 'M'
4      fois la lettre 'N'
3      fois la lettre 'O'
2      fois la lettre 'P'
3      fois la lettre 'R'
3      fois la lettre 'S'
1      fois la lettre 'T'
1      fois la lettre 'X'
```

**20-**

**Soit le code suivant :**

```
#include <stdio.h>
#include <string.h>
main()
{
    /* Déclarations */
    char CH1[101], CH2[101]; /* chaînes données */
    char *P1, *P2; /* pointeurs d'aide dans CH1 et CH2 */
    int TROUVE; /* indicateur logique: vrai, si le caractère */
                /* actuel dans CH1 a été trouvé dans CH2.    */

    /* Saisie des données */
```

```

printf("Entrez la première chaîne de caractères"
      " (max.100 caractères) :\n");
gets(CH1);
printf("Entrez la deuxième chaîne de caractères"
      " (max.100 caractères) :\n");
gets(CH2);
/* Eliminer les lettres communes */
/* Idée: Parcourir CH2 de gauche à droite et contrôler */
/* pour chaque caractère s'il se trouve aussi dans CH1. */
/* Si tel est le cas, éliminer le caractère de CH1 à */
/* l'aide de strcpy. */
for (P2=CH2; *P2; P2++)
{
    TROUVE = 0;
    for (P1=CH1 ; *P1 && !TROUVE ; P1++)
        if (*P2==*P1)
        {
            TROUVE = 1;
            strcpy(P1, P1+1);
        }
}
/* Affichage du résultat */
printf("Chaîne résultat : \"%s\" \n", CH1);
return 0;
}

```

Réponse :

**Programme qui lit deux chaînes de caractères CH1 et CH2 au clavier et supprime la première occurrence de CH2 dans CH1.**

```

Entrez la première chaîne de caractères (max.100 caractères) :
correction d'examen session principale
Entrez la deuxième chaîne de caractères (max.100 caractères) :
d'examen
Chaîne résultat : "corrctio en session principale"
-----

```

**21-**

Soit le code suivant :

```

#include <stdio.h>

main()
{
    int i,j;

    int t1[10]={1,3,3,-6,88,77,-77,8,7,8};

```

```

int t2[10]={1,-2,3,-5,8,9,-1,7,5,9};

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;

for(i=0;i<10;i++)

{

printf("%d |",t1[i]);

}

printf("\n");

for(i=0;i<10;i++)

{

printf("%d |",t2[i]);

}

return 0;}

```

Réponse :

un programme permettant de copier les éléments positifs d'un tableau t2 dans un deuxième tableau t1  
Mettre des zéros dans les cases vides si les eles positifs de t2 < 10 ..

```

1 |3 |8 |9 |7 |5 |9 |0 |0 |0 |
1 |-2 |3 |-5 |8 |9 |-1 |7 |5 |9 |
-----
Process exited after 0.1921 seconds with return value 0
Appuyez sur une touche pour continuer...

```