

Série 2 programmation- Langage C
Les structures alternatives et répétitives
1^{ère} année Ingénieur Informatique

Exercice 1 :

Sans taper les programmes suivants sur machine, donner le résultat de leurs exécutions

Programme 1

```
#include <stdio.h>
void main()
{
    int i=1;
    while (i<3)
    {
        printf("%d \n", i);
        i++;
    }
    printf("après la boucle, la valeur de i est %d \n",i);
}
```

Programme 2

```
#include <stdio.h>
void main()
{
    int i;
    for(i=1; i<3; i++)
        printf(" %d \n",i);
    printf("la valeur de i est %d \n",i);
}
```

Programme 3

```
#include <stdio.h>
void main()
{
    int i;
    for(i=1; i<3; i++)
    {
        printf(" %d \n",i);
        printf("la valeur de i est %d \n",i);
    }
}
```

Programme 4

```
#include <stdio.h>
void main()
{
    int i,j;
    for(i=1; i<3; i++)
        for(j=1; j<4; j++)
            printf("i : %d, j : %d \n",i,j);
}
```

Programme 5

```
#include <stdio.h>
void main()
{
    int i,j;
    for(i=1; i<3; i++)
        for(j=1; j<4; j++)
            printf("i : %d, j : %d \n",i,j);
            printf("les valeurs de i et j sont %d, %d \n", i, j);
}
```

Programme 6

```
#include <stdio.h>
void main()
{
    int i,j;
    for(i=1; i<3; i++)
    {
        for(j=1; j<4; j++)
            printf("i : %d, j : %d \n",i,j);
            printf("les valeurs de i et j sont %d, %d \n", i, j);
    }
}
```

Programme 7

```
#include <stdio.h>
void main()
{
    int i,j;
    for(i=1; i<3; i++)
    {
        for(j=1; j<4; j++)
        {
            printf("i : %d, j : %d \n",i,j);
            printf("les valeurs de i et j sont %d, %d \n", i, j);
        }
    }
}
```

Programme 8

```
#include <stdio.h>
void main()
{
    int i,j;
    for(i=1; i<3; i++)
    {
        for(j=1; j<4; j++)
            printf("i : %d, j : %d \n",i,j);
            printf("la valeur de j est %d \n",j);
    }
    printf("la valeur de i est %d \n",i);
}
```

Programme 9

```
#include <stdio.h>
void main()
{
    int i,j,k;
    for (i=0;i<1; i++)
    {
        for(j=0; j<2;j++)
        {
            for (k=0;k<3; k++)
                printf("\n %d", k+10*j+100*i);
            printf("\n i:%d, j:%d, k:%d",i,j,k);
        }
        printf("\n i:%d, j:%d, k:%d",i,j,k);
    }
    printf("\n i:%d, j:%d, k:%d \n",i,j,k);
}
```

Réécrire ce même programme en remplaçant for par while

Programme 10

```
#include<stdio.h>
void main()
{
    int i,j,k;
    for (i=1;i<3; i++)
        for (j=1;j<3; j++)
            printf("%d \n", i+j);
    for (k=1; k<3; k++)
        printf("%d \n", i+j+k);
}
```

Réécrire ce même programme en remplaçant for par while

Exercice 2: (Extrait du DS de programmation du 06 novembre 2012)

1. Si on exécute ces deux programmes, qu'est ce qu'ils affichent?

Programme1

```
void main()
{
    int i,j;
    for(i=0; i<2; i++)
    {
        for (j=0; j<2; j++)
            printf("\n i:%d, et j:%d", i,j);
        printf("\n i:%d, et j:%d", i,j);
    }
    printf("\n i:%d, et j:%d", i,j);
}
```

Programme 2

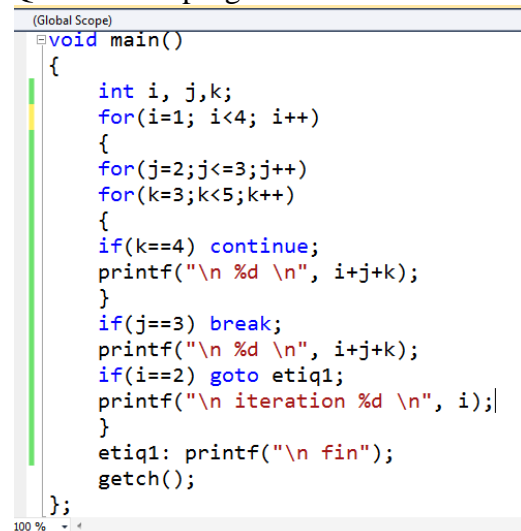
```
void main()
{
    int i,j;
    for(i=0; i<3; i++)
    {
        if (i==1) continue;
        for (j=0; j<4; j++)
        {
            printf("\n i:%d, et j:%d", i,j);
            if(j==2) break;
        }
        printf("\n i:%d, et j:%d", i,j);
    }
    printf("\n i:%d, et j:%d", i,j);
}
```

2. Pour le programme 2, remplacer les 2 boucles for par 2 boucles while

Exercice 3:

(Extrait du DS de programmation du 12 novembre 2013)

Qu'affiche le programme suivant?



```
void main()
{
    int i, j, k;
    for(i=1; i<4; i++)
    {
        for(j=2; j<=3; j++)
        {
            for(k=3; k<5; k++)
            {
                if(k==4) continue;
                printf("\n %d \n", i+j+k);
            }
            if(j==3) break;
            printf("\n %d \n", i+j+k);
            if(i==2) goto etiq1;
            printf("\n iteration %d \n", i);
        }
        etiq1: printf("\n fin");
        getch();
    }
};
```

(extrait du DS du 11 novembre 2014): Qu'affiche le programme suivant?

```
void main()
{
    int i,j,k;
    for(i=1;i<3;i++)
    {
        for(j=2;j<5;j++)
        {
            if(j==3) continue;
            printf("\n %d", i*10+j);
            if(j==4) break;
        }
        for(j=2;j<5;j++)
        {
            if(j==3) continue;
            printf("\n %d", i*10+j);
            if(j==4) break;
        }
        if(i==2) goto FIN;
    }
    printf("\n devoir surveille ");
    FIN: printf("\n FIN ");  getch(); }
```

Affichage (9 lignes)

```
.....
.....
.....
.....
.....
.....
.....
.....
.....
```

(extrait du DS du 11 novembre 2015)

Qu'affiche le programme suivant:

```
(Global Scope)
void main()
{
    int i,j,k;
    for(i=0; i<4;i++)
    {
        for(j=0;j<4;j++)
        {
            if(j==2) continue;
            printf("\n %d ",i+j);
            for(k=3;k<5;k++)
            printf("\n %d ",i+j+k);
            if(j==3) goto e1;
        }
        if(i==3)break;
    }
    printf("\n fin des boucles ");
    e1: printf("\n FIN ");
    getch();
}
```

10 lignes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Extrait du DS du 08 novembre 2016

Qu'affiche le programme suivant?

```
(Global Scope)
void main()
{
    int i,j=1,k;
    for(i=10;i<13; i++)
    do
    {
        for(k=2; k<7; k++)
        {
            if(i==11) continue;
            printf("\n %d ", i+j+k);
            if(j==2) break;
            if(k==5) goto val;
        }
        j++;
    } while(j<3);
    printf("\n fin des boucles \n");
    val: printf("\n FIN ");
    getch();
}
```

// 5 lignes

.....

.....

.....

.....

.....

Extrait du DS de novembre 2017

```
(Global Scope)
void main()
{
    int i=1,j,k,m=2;
    do
    {
        for( ; i<5; i+=2 )
            for( k=3; k<6 ; k+=2 )
                printf("%d\n",i+k+m);
        if(m==4) goto etiquette;
        for( j=2; j<10 ; j+=3 )
        {
            printf("%d\n",i+j+k+m);
            if(j==5) continue;
            if(m==3) break;
        }
        m++;
    }
    while( m<5 );
    printf("\n fin des boucles ");
    etiquette: printf("\n fin du travail ");
    getch();
}
```

// 9 lignes

// 9 lignes

.....

.....

.....

.....

.....

.....

.....

.....

.....

Extrait du DS d'octobre 2018 (affichage 10 lignes)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,k;
    for(i=1; i<5; i+=2)
    {
        j=1;
        do
        {
            if(j==2 ) continue;
            k=5;
            while(k>0)
            {
                printf("\n %d ", i+j+k);
                k-=3;
            }
            j+=2;
        }
        while(j<5);
        if( i==3) goto etiquette;
        printf("\n %d ", i+j+k);
    }
    printf("\n Fin DS");
    etiquette: printf("\n FIN ");
    getch();
}
```

Exercice 4 :

Écrire un programme qui lit deux entiers (A et B) au clavier et qui affiche le signe du produit de A et B sans faire la multiplication.

Exercice 5 :

Afficher un triangle isocèle formé d'étoiles de n lignes (n est fourni au clavier):

Nombre de lignes : 8

```

*
***
*****
*****
*****
*****
*****
*****
*****
```

Exercice 6 :

Ecrire un programme qui résout l'équation du second degré $ax^2+bx+c=0$. (a, b et c sont saisis au clavier).

Exercice 7 :

Ecrire un programme qui affiche au choix (selon le caractère à saisir au clavier +, ou *, ou /) la table d'addition, ou de multiplication ou de division. (Soigner l'affichage)

Exercice 8 :

Calculer la factorielle d'un entier naturel N en utilisant **while**, puis en utilisant **for**.

Exercice 9 :

Écrire un programme qui lit N nombres entiers au clavier et qui affiche leur somme, leur produit et leur moyenne. Le nombre N est à entrer au clavier.

- en utilisant **while**,
- en utilisant **do - while**,
- en utilisant **for**.

Exercice 10 :

Ecrire un programme qui lit un caractère au clavier (utiliser getchar), et le convertit en minuscule s'il est majuscule, et inversement.

Exercice 11:

Ecrire un programme qui lit un texte, se terminant par #, formé par des 0 et des 1 (utiliser getchar), ensuite remplace le 1 par 0 et inversement. Si le caractère est différent de 0 ou de 1, il reste inchangé.

Exemple :

```
1a00011z01#
0a11100z10
```

Exercice 12 :

Ecrire un programme qui lit un texte se terminant par # (on suppose que les caractères sont des 0 et des 1 seulement ; utiliser getchar), ensuite effectue une compression selon l'approche suivante : Chaque suite de n caractères identiques est remplacée par : $n^{\text{le caractère}}$

Exemple :

```
111000001111000000#
3^1#5^0#4^1#6^0#
```

Exercice 13 :

Ecrire un programme qui accepte un texte qui se termine par # (utiliser getchar), et affiche sur chaque ligne un mot du texte.

Exemple : azerty qsd fg wxcvb #

```
azerty
qsd fg
wxcvb
```

Exercice 14 :

Écrire un programme qui lit un texte du clavier qui se termine par # (utiliser getchar), et affiche:

- Nombre total de caractères

- Nombre de lignes (\n)
- Nombre de séparateurs (, . ; : \n)
- Nombre de mots
- Nombre de mots dont la longueur est >4
- Longueur du mot le plus long
- Nombre de mots formés par des caractères identiques.

NB : Le texte peut commencer par un ou plusieurs séparateurs (' , . : ; \n).

Les mots peuvent être séparés par un ou plusieurs séparateurs.

Le texte peut s'étendre sur plusieurs lignes.

Exercice 15 :

Ecrire un programme qui accepte un texte qui se termine par # (utiliser getchar), et affiche le nombre de mots formés par des caractères alphabétiques, et le nombre de mots formés par des chiffres.

Exemple :

Azer ghj234 567 8765 der4565 9876#

Le nombre de mots alphabétiques est 1, le nombre de mots chiffres est 3

Exercice 16: (Extrait du DS de programmation du 06 novembre 2012)

Ecrire un programme C qui lit un texte se terminant par # (utiliser getchar), et affiche le nombre de mots contenant au moins 5 voyelles (A, E, I, O, U, Y)

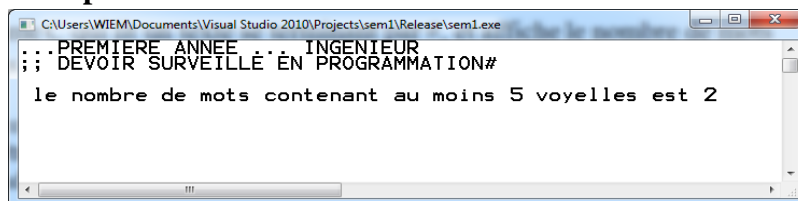
NB: Le texte peut commencer par un ou plusieurs séparateurs (' , . : ;).

Les mots peuvent être séparés par un ou plusieurs séparateurs.

Le texte peut s'étendre sur plusieurs lignes.

Ne pas utiliser de tableaux

Exemple:



Exercice 17: (Extrait du DS de programmation du 12 novembre 2013)

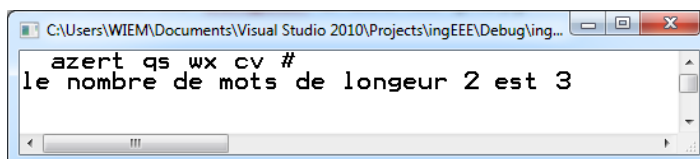
Écrire un programme qui lit un texte du clavier qui se termine par # (utiliser getchar), et affiche le nombre de mots dont la longueur est égale à 2.

NB : Le texte peut commencer par un ou plusieurs séparateurs (' , . : ; \n).

Les mots peuvent être séparés par un ou plusieurs séparateurs.

Le texte peut s'étendre sur plusieurs lignes.

Ne pas utiliser les tableaux.



Exercice 18: extrait du DS de programmation du 14 novembre 2014)

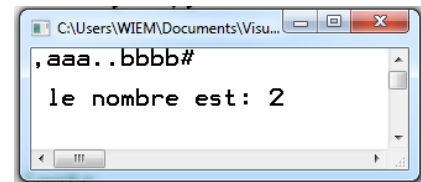
Ecrire un programme C qui lit un texte se terminant par # (utiliser getchar), et affiche le nombre de mots contenant au moins une séquence successive du même caractère 3 fois.

NB: Le texte peut commencer par un ou plusieurs séparateurs (' , . : ;).

Les mots peuvent être séparés par un ou plusieurs séparateurs.

Le texte peut s'étendre sur plusieurs lignes.

Ne pas utiliser de tableaux



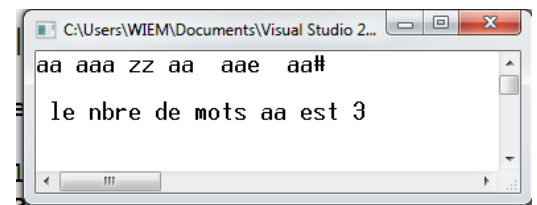
Exercice 19: extrait du DS de programmation du 03 novembre 2015

Ecrire un programme qui lit un texte au clavier (caractère par caractère: utiliser getchar), et affiche le nombre de mots aa.

Le texte peut commencer par un ou plusieurs séparateurs (',' ou '.' ou ';' ou ' ')

Les mots peuvent être séparés par un ou plusieurs séparateurs

Ne pas utiliser les tableaux



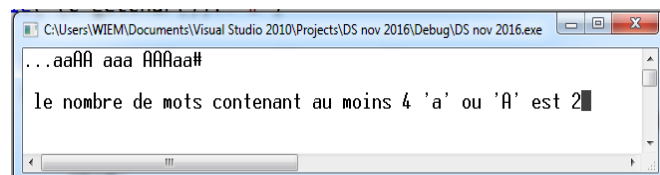
Exercice 20: extrait du DS de programmation du 08 novembre 2016

Ecrire un programme qui lit un texte au clavier (caractère par caractère: utiliser getchar), et affiche le nombre de mots contenant au moins 4 fois la lettre 'a' ou 'A'.

Le texte peut commencer par un ou plusieurs séparateurs (',' ou '.' ou ';' ou ' ')

Les mots peuvent être séparés par un ou plusieurs séparateurs

Ne pas utiliser les tableaux



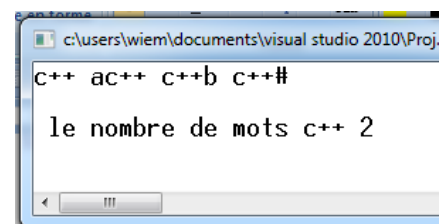
Exercice 21: extrait du DS de novembre 2017

Ecrire un programme qui lit un texte au clavier (caractère par caractère: utiliser getchar), et affiche le nombre de mots c++.

Le texte peut commencer par un ou plusieurs séparateurs (',' ou '.' ou ';' ou ' ')

Les mots peuvent être séparés par un ou plusieurs séparateurs

Ne pas utiliser les tableaux



Exercice 22: extrait du DS de programmation octobre 2018

Ne pas utiliser les tableaux, commencer directement par void main.

Ecrire un programme qui lit un texte au clavier (caractère par caractère: utiliser getch), et affiche le nombre de mots formés par des lettres majuscules.

Le texte peut commencer par un ou plusieurs séparateurs (',' ou '.' ou ';' ou ' ')

Les mots peuvent être séparés par un ou plusieurs séparateurs

