

2

Prise en main

```
#include <stdio.h>

int main() {
    printf("Hello world\n");
    return 0;
}
```

3

Variables

```
#include <stdio.h>

int main() {
    printf("Taille d'un char : %lu\n", sizeof(char));
    printf("Taille d'un unsigned char : %lu\n", sizeof(unsigned char));
    printf("Taille d'un short : %lu\n", sizeof(short));
    printf("Taille d'un unsigned short : %lu\n", sizeof(unsigned short));
    printf("Taille d'un int : %lu\n", sizeof(int));
    printf("Taille d'un unsigned int : %lu\n", sizeof(unsigned int));
    printf("Taille d'un long int : %lu\n", sizeof(long int));
    printf("Taille d'un long long int : %lu\n", sizeof(long long int));
    printf("Taille d'un float : %lu\n", sizeof(float));
    printf("Taille d'un double : %lu\n", sizeof(double));
    printf("Taille d'un long double : %lu\n", sizeof(long double));
}
```

```
#include <stdio.h>

int main() {
    char c = 'a';
    unsigned char uc = 0;
    short s = -1;
    unsigned short us=65535;
    int i = 0;
    unsigned int ui = 0;
    long int li = 0;
    float f = 0.5;
    double d = 1e10;
    long double ld = 5e-3;

    printf("char\t\t: valeur=%c, taille=%ld\n", c, sizeof(c));
    printf("unsigned char\t\t: valeur=%d, taille=%ld\n", uc, sizeof(uc));
    printf("short\t\t: valeur=%d, taille=%ld\n", c, sizeof(s));
}
```

```

printf("unsigned short\t\t: valeur=%d, taille=%ld\n", c, sizeof(us));
printf("int\t\t: valeur=%d, taille=%ld\n", c, sizeof(i));
printf("unsigned int\t\t: valeur=%d, taille=%ld\n", c, sizeof(ui));
printf("long int\t\t: valeur=%u, taille=%ld\n", c, sizeof(li));
printf("float\t\t: valeur=%ld, taille=%ld\n", c, sizeof(f));
printf("double\t\t: valeur=%f, taille=%ld\n", c, sizeof(d));
printf("long double\t\t: valeur=%f, taille=%ld\n", c, sizeof(ld));
}

```

4

Fonctions

```

#include <stdio.h>

int suivant(int i) {
    int j = i+1;
    return j;
}

int main() {
    int k = 3, j;

    j = suivant(k);
    printf("Le nombre suivant de %d est %d\n", k, j);
    return 0;
}

```

```

#include <stdio.h>

float somme(unsigned int a, short b, float c) {
    return a+b+c;
}

int main() {
    unsigned int v1 = 70000;
    short v2 = 16;
    float v3 = 0.5;

    printf("La somme de %u, %d et %f est %f\n", v1, v2, v3, somme(v1, v2, v3));
    return 0;
}

```

```

#include <stdio.h>

void message(char t[], char s[]) {
    printf("Message (%s) : %s\n", t, s);
}

```

```

    return; // optionnel
}

int main() {
    message("Erreur", "Ceci est une erreur grave!!");
    message("Avertissement", "Bon je vais laisser courir pour cette fois.");
    return 0;
}

```

sommeProduit n°1

```

#include <stdio.h>

void sommeProduit(int i, int j) {
    int k=i * j; // stockage temporaire de la valeur qui ira dans j
    i = i+j; // i reçoit la somme des paramètres
    j = k; // j reçoit le produit des paramètres (stocké dans k)
    return;
}

int main() {
    int i = 3; // déclaration de i et initialisation à 3
    int j = 4; // déclaration de j et initialisation à 4;
    printf("i=%d ; j=%d\n", i, j); // affiche : i=3 : j=4
    sommeProduit(i,j); // appel de la fonction : on pourrait s'attendre a ce que
    // i vaille 7 et j vaille 12 en sortie
    printf("i=%d ; j=%d\n", i, j); // affiche encore: i=3 ; j=4
    return 0;
}

```

sommeProduit n°2

```

#include <stdio.h>

void sommeProduit(int i, int j) {
    int k=(*i) * (*j); // stockage temporaire de la valeur qui ira dans *j
    *i = *i + *j; // *i reçoit la somme des paramètres
    *j = k; // *j reçoit le produit des paramètres (stocké dans k)
    return;
}

int main() {
    int i = 3; // déclaration de i et initialisation à 3
    int j = 4; // déclaration de j et initialisation à 4;
    printf("i=%d ; j=%d\n", i, j); // affiche : i=3 : j=4
    sommeProduit(i,j); // appel de la fonction : on pourrait s'attendre a ce que
    // i vaille 7 et j vaille 12 en sortie
    printf("i=%d ; j=%d\n", i, j); // affiche encore: i=7 ; j=12
    return 0;
}

```

5

```
#include <stdio.h>

int main() {
    printf("Youpi!\n");
    printf("Tralala\n");
    return 0;
}
```

```
#include <stdio.h>

int main() {
    int i=3;
    if (i%2 == 0) {
        printf("Youpi\n");
    }
    printf("Tralala\n");
    return 0;
}
```

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

int main() {
    srand(time(NULL));

    if(rand() % 2 == 0) {
        printf("Youpi!\n");
    }
    printf("Tralala\n");
    return 0;
}
```

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

int main() {
    srand(time(NULL));
    int i=rand();
    if (i % 2 == 0){
        printf("J'ai le nombre pair %d -> ", i);
        printf("Youpi!\n");
    }
}
```

```

    }
    else {
        printf("J'ai le nombre impair %d -> ", i);
        printf("Tralala\n");
    }
    printf("Ceci est toujours execute\n");
    return 0;
}

```

6

```

#include <stdio.h>

int main() {
    int i=0;
    while (i < 20) {
        printf("%02de instruction de la boucle\n", i);
        i++;
    }
    return 0;
}

```

```

#include <stdio.h>

int main() {
    int i=0;
    i=0;
    while (i < 0) {
        printf("(1) %02de instruction de la boucle\n", i);
    }
    i=0;
    do {
        printf("(2) %02de instruction de la boucle\n", i);
    } while (i < 0);
    return 0;
}

```

Avant modification

```

#include <stdio.h>

int main() {
    int i=0,n;
    do {
        printf("Saisir n, 1<=n<=20: ");
        scanf("%d",&n);
        i++;
    }
}

```

```
    } while (n<1 || n>20);  
    printf("n=%d, apres %d Itérations. \n", n, i);  
    return 0;  
}
```

Après modification

```
#include <stdio.h>  
  
int main() {  
    int i=0,n;  
    do {  
        printf("Saisir n, n<=0: ");  
        scanf("%d",&n);  
        i++;  
    } while (n > 0);  
    printf("n=%d, apres %d Itérations. \n", n, i);  
    return 0;  
}
```

```
#include <stdio.h>  
  
int main() {  
    int i;  
    i=0;  
    while (i <10) {  
        printf("Valeur : %d\n", i);  
        i++;  
    }  
    return 0;  
}
```

```
#include <stdio.h>  
  
int main() {  
    float tab[100];  
    int i;  
    for (i=0; i <100; i++) {  
        tab[i] = (float) i/100;  
        printf("Valeur insere en position %d : %f\n", i, tab[i]);  
    }  
    return 0;  
}
```

Tableaux

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

```
#include <time.h>

float aleat () {
    return (float)rand() / RAND_MAX;
}

int main() {
    srand(time(NULL)); // initialisation du generateur de nombres aleatoires
    int N = rand()%200+1; // generation d'un nombre aleatoire entre 1 et 200 inclus
    float tab[N]; // declaration d'un tableau de N float
    // remplissage du tableau avec des nombres entre 0 et 1
    int i;
    for (i=0; i<N; i++)
        tab[i] = aleat ();
    // calcul de la moyenne des nombres dans le tableau
    float m = 0;
    for (i=0; i<N; i++)
        m += tab[i];
    m /= N;
    printf("La moyenne est: %f\n", m);
    // recherche du premier element plus grand que 0.9
    for (i=0; i < N && tab[i] < 0.9; i++);
    if (i < N)
        printf("Élément trouvé %f en position_%d\n", tab[i], i);
    else
        printf("Aucun élément trouvé \n");
    return 0;
}
```