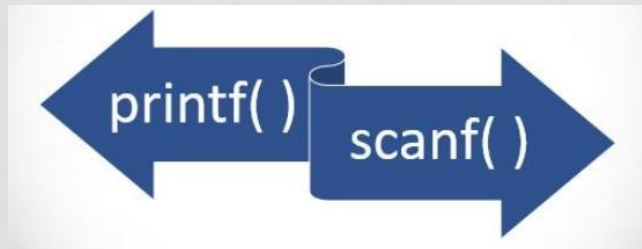


LA PROGRAMMATION EN LANGAGE C

LES ENTRÉES SORTIES SÉANCE 3 : LECTURE ÉCRITURE

SMI/ S3



Pr S.EL FILALI
sanaa.elfilali@etu.univh2c.ma

SOMMAIRE

- Affichage / Ecriture /sortie : printf()
- Affichage d'un texte
- Affichage d'une variable
- Spécificateurs de Format pour printf()
- Affichage multiple
- Exemples / Exercices
- Les Opérateurs
- Les Opérateurs arithmétiques
- Les Opérateurs de comparaison
- Les opérateurs logique
- Les opérateurs de manipulation de bits
- Les opérateurs d'affectation
- Les Opérateurs abrégés (opérateurs d'assignement)
- La priorité des opérateurs
- La fonction scanf() : lecture /entrée
- Exemples / Exercices

AFFICHAGE / ECRITURE PRINTF()

La fonction printf()

- C'est une instruction d'écriture
- Elle permet à un programme de communiquer des informations sous forme de résultats par l'intermédiaire d'un périphérique : écran.
- Printf une fonction de la bibliothèque `stdio.h`

AFFICHAGE D UN TEXTE

```
printf("BONJOUR");
```

pas de retour à la ligne du curseur après l'affichage,

```
printf("BONJOUR\n");
```

affichage du texte, puis retour à la ligne du curseur.

AFFICHAGE D'UNE VARIABLE

Fonction de la bibliothèque `stdio.h`

```
printf("%format",variable);
```

format :

d

- pour les décimaux,

x

- pour les hexadécimaux,

c

- pour un caractère, etc.

AFFICHAGE D'UNE VARIABLE

Fonction de la bibliothèque `stdio.h`

Format : `printf("%format",variable);`

Exemples :

```
printf("bonjour");
```

```
printf("la valeur est : %d ",i);
```

```
printf("les valeurs sont : %d et %d",i,j);
```

SPÉCIFICATEURS DE FORMAT POUR PRINTF()

SYMBOLE	TYPE	IMPRESSION COMME
%d ou %i	int	Entier relatif
%u	int	Entier naturel (unsigned)
%o	int	Entier exprimé en octal
%x	Int	Entier exprimé en hexadécimal
%c	int	caractère
%f	double	rationnel en notation décimale
%e	double	rationnel en notation scientifique
%s	char*	chaîne de caractères

AFFICHAGE MULTIPLE

```
printf("format1.... formatN",variable1,.....,variableN);
```


EXEMPLE PRINTF

Exemple 1:

```
Int q=100;  
Float p = 60.50;  
Char t = 'A';  
printf("quantite = %d \n ",q);  
printf(" Prix= %f\n ",p);  
printf(" Taille= %c \n ",t);  
printf(" Prix Total =%f \n ",q*p);
```

ou bien

```
printf("Quantite = %d\nPrix = %f\nTaille = %c\nPrix total = %f",q,p,t,q*p);
```

Imprime :

```
Quantite = 100  
Prix = 60.500000  
Taille = A  
Prix total = 6050,000000
```

EXAMPLE PRINTF

Exemple 2:

float val = 25.1234;

<code>printf("%f",val);</code>		<i>/* affiche : 25.123400 */</i>
<code>printf("%.0f",val);</code>		<i>/* affiche : 25 */</i>
<code>printf("%.1f",val);</code>		<i>/* affiche : 25.1 */</i>
<code>printf("%.2f",val);</code>		<i>/* affiche : 25.12 */</i>
<code>printf("%e",val);</code>		<i>/* affiche : 2.512340e+01 */</i>
<code>printf("%.3e",val);</code>		<i>/* affiche : 2.512e+01 */</i>

EXERCICE1.2

- z Ecrire un programme C permettant de créer 2 variables
2 caractères initialisés à 65 et 'A'
- z Afficher ces variables sous forme d'entiers et de caractères

SOLUTION EXERCICE 1.2

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

void main()
{
    char u,v;
    u=65;
    v='A';
    printf("u = %d, %c",u,u);
    printf("v= %d, %c",v,v);
    getch();
}
```

LES OPÉRATEURS

Il existe 4 types d'opérateurs:

Arithmétiques

Logiques

Relationnels

Assignment.

OPÉRATEURS ARITHMÉTIQUES

+

- addition

-

- soustraction

*

- multiplication

/

- division (entière et rationnelle!)

%

- modulo (reste d'une div. entière)

EXERCICE 1.3

Ecrire un programme C déclarant 2 variables entières et affichant leur somme multipliée par 2.

SOLUTION EXERCICE 1.3

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int a,b,somme;
    a=10;
    b=50;
    somme=(a+b)*2;
    printf("résultat : %d\n",somme);
    printf("Appuyer sur une touche\n");
    getch();
}
```


OPÉRATEURS DE COMPARAISON

Ces opérateurs produisent une valeur booléenne vraie (**true**) ou fausse (**false**).

==

- égal à

!=

- différent de

<, <=, >,
>=

- plus petit que, ...

OPÉRATEURS LOGIQUE

&& et logique (**and**)

|| ou logique (**or**)

! négation logique (**not**)

OPÉRATEURS LOGIQUE

Les résultats des opérations de comparaison et des opérateurs logiques sont du type int:

- la valeur **1** correspond à la valeur booléenne vrai
- la valeur **0** correspond à la valeur booléenne faux

OPÉRATEURS LOGIQUE

Les opérateurs logiques considèrent :

toute valeur **différente de zéro** comme vrai

Et toute **valeur zéro** comme faux.

OPÉRATEURS DE MANIPULATION DE BITS

Fonction	Symbole	Remarques
ET	&	$c = a \& b$; \rightarrow si $a = 1001\ 0011$ et $b = 0101\ 0110$, $c = 0001\ 0010$
OU		$c = a b$; \rightarrow si $a = 1001\ 0011$ et $b = 0101\ 0110$, $c = 1101\ 0111$
OU exclusif	^	$c = a \wedge b$; \rightarrow si $a = 1001\ 0011$ et $b = 0101\ 0110$, $c = 1100\ 0101$
Décalage à droite	>>	$(a >> 2)$ décale la valeur de a de 2 bits vers la droite
Décalage à gauche	<<	$(a << 2)$ décale la valeur de a de 2 bits vers la gauche
Complément à un	~	$c = \sim a \rightarrow$ si $a = 1001\ 0011$, $c = 0110\ 1100$

```
unsigned char u_c = 1;
```

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

```
u_c = u_c << 1;
```

0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---

```
u_c = u_c << 2;
```

0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

```
u_c = u_c >> 3;
```

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

LES OPÉRATEURS D'AFFECTATION

En pratique, nous retrouvons souvent des affectations comme:

$$i = i + 2$$

En C, nous utiliserons plutôt la formulation plus compacte:

$$i += 2$$

L'opérateur `+=` est un *opérateur d'affectation*.

LES OPÉRATEURS D'AFFECTATION

Somme = somme + i;

Ici, la valeur de la variable **somme** est additionnée à **i** puis réassignée (ensuite !) à la même variable **somme**.

LES OPÉRATEURS D'AFFECTATION

Pour la plupart des expressions de la forme:

$$\text{expr1} = (\text{expr1}) \text{ op } (\text{expr2})$$

$$\text{expr1} \text{ op} = \text{expr2}$$

+=

ajouter à

-=

diminuer de

***=**

multiplier par

/=

diviser par

%=

modulo

LES OPÉRATEURS ABRÉGÉS (OPÉRATEURS D'ASSIGNEMENT)

opérateur abrégé	Expression abrégée	Expression équivalente
+=	c += 3;	c = c + 3;
-=	c -= 3;	c = c - 3;
*=	c *= 3;	c = c * 3;
/=	c /= 3;	c = c / 3;
%=	c %= 3;	c = c % 3;

PRIORITÉ DES OPÉRATEURS

N priorité	Type d'opérateur	Opérateurs	Associativité
15	Primaire	() [] . ->	g à d
14	Unaire	++ -- (nom Type) * & sizeof + - ! ~	g à d
13	Arithmétique	* / %	g à d
12	Arithmétique	+ -	g à d
11	De manipulation de bits	>> <<	g à d
10	De relation	> < >= <=	g à d
9	De relation	== !=	g à d
8	De manipulation de bits	&	g à d
7	De manipulation de bits	^	g à d
6	De manipulation de bits		g à d
5	Logique	&&	g à d
4	Logique		g à d
3	Conditionnel	?:	g à d
2	D'affectation	= += -= *= /= =	g à d

LA FONCTION SCANF()

Définition :

La fonction scanf() :

- appartenant à la bibliothèque **stdio.h**
- permet de lire des données au clavier et de les affecter à des variables.
- scanf (scan : formatted) peut être utilisée pour la saisie des données formatées (lettre, chiffre ou chaîne de caractère)
C'est une fonction variadique (peut prendre un nombre variable de paramètres)
La fonction retourne le nombre de variable affectées par la saisie
=> Permet de vérifier si la saisie s'est bien passée

LA FONCTION SCANF()

Syntaxe :

```
scanf("format_variable1 format_variable 2 ....format_variable n", &variable_1,  
      &variable_2, ..., & variable_n);
```

LA FONCTION SCANF() EXEMPLE

```
char alpha;
```

```
int i;
```

```
float r;
```

```
scanf("%c",&alpha);  $\Longrightarrow$  /* saisie d'un caractère */
```

```
scanf("%d",&i);  $\Longrightarrow$  /* saisie d'un nombre entier */
```

```
scanf("%x",&i);  $\Longrightarrow$  /* saisie d'un nombre entier en hexadécimal */
```

```
scanf("%f",&r);  $\Longrightarrow$  /* saisie d'un nombre réel */
```

LA FONCTION SCANF() EXEMPLE

```
#include <stdio.h>
#include <string.h>
void main()
{
    // declaration des variables
    int V1,V2,V3;
    char S1[30], S2[30];
    int Somme ;
    // lecture des données
    printf("votre nom : "); scanf("%s", S1);
    printf("votre prenom : " ) ; scanf("%s", S2);

    // lecture et traitement
    printf("la saisie des donnees :\n ");
    Somme=scanf("%d%d%d",&V1,&V2,&V3);
    //affichage

    printf("%s %s vous avez saisi % d nombres ",S1, S2, Somme);

}
```

```
votre nom : ELFILALI
votre prenom : SANAA
la saisie des donnees :
3
4
5
ELFILALI SANAA vous avez saisi 3 nombres
```

AIRE D UN CERCLE

```
/* Premier programme en C */  
#include <stdio.h>  
void main()  
{  
    float aire, pi, rayon;  
    // rayon = 3.7;  
    printf("Entrez une valeur pour rayon : ");  
    scanf("%f", &rayon);  
    pi = 3.14159;  
    aire = pi*rayon*rayon;  
    printf("Aire est %f\n", aire);  
}
```

AIRE D'UN CERCLE

```
printf("Entrez une valeur pour rayon : ");
```

```
// Ce printf n'a pas de \n (nouvelle ligne)
```

```
scanf("%f", &rayon);
```

```
//scanf prend la valeur réelle entrée (%f) et l'assigne à la  
variable rayon.
```


AIRE D'UN CERCLE (AMÉLIORATION) EXÉCUTION DU PROGRAMME

Entrez une valeur pour rayon : 3.7

Aire est 43.008369

LA FONCTION SCANF() EXEMPLE

Saisie de plusieurs variables.

```
{
```

```
float a, b, c, det ;
```

```
printf("Donnez les valeurs de a,b et c : ");
```

```
/* Saisie de a, b, c */
```

```
scanf("%f %f %f",&a,&b,&c);
```

```
/* Calcul du déterminant */
```

```
det = b*b- 4*a*c;
```

```
printf("Le déterminant = %.2f", det);
```

```
}
```

LA FONCTION SCANF() EXEMPLE

Tampon de scanf

Les informations tapées au clavier sont d'abord mémorisées dans un emplacement mémoire appelé *tampon* (**buffer**).

elles sont mises à la disposition de **scanf** après l'activation de la touche <Entrée>

➤ Pour les nombres :

scanf avance jusqu'au premier caractère différent d'un séparateur (espace, tabulation, retour à la ligne)

puis **scanf** prend en compte tous les caractères jusqu'à la rencontre d'un séparateur ou d'un caractère invalide ou en fonction du largeur.

➤ Pour les caractères %c :

- **scanf** prend le caractère courant qu'il soit séparateur ou non séparateur.

LA FONCTION SCANF() EXEMPLE

Tampon de scanf : résoudre le problème de la saisie des caractères

➤ Pour les caractères %c : il faut laisser un espace avant le % : exemple
`scanf(" %c",&ch);`

Un espace



CONVERTIR LE CARACTÈRE MAJUSCULE EN MINUSCULE ET MINUSCULE EN MAJUSCULE

Exercice :

écrire un programme qui permet de convertir un caractère saisi au clavier en majuscule et minuscule

CONVERTIR LE CARACTÈRE MAJUSCULE EN MINUSCULE ET MINUSCULE EN MAJUSCULE

```
#include <stdio.h>
void main(void)
{
    char ch;

    printf("veuillez saisir un caractere : "); scanf(" %c",&ch);

    printf("caractere saisi est : %c\t \n
           en maj = %c \t en min = %c\n", ch,toupper(ch), tolower(ch));
}
```

```
veuillez saisir un caractere : S
caractere saisi est : S
    en maj = S      en min = s
```

CONVERTIR LE CARACTÈRE MAJUSCULE EN MINUSCULE ET MINUSCULE EN MAJUSCULE

```
#include <stdio.h>
#include <ctype.h>
#include <time.h>
void main(void)
{

    char ch;
    time_t    t = time(NULL);
    printf(" \t exemple cours : Exercice conversion      %s\n", asctime(localtime(&t)));
    printf(" \t ===== \n\n");
    printf("*****\n");
    printf("Realise par : SMI S3 \n");
    printf("Encadre par Professeur: S.ELFILALI\n");
    printf("*****\n");
    printf("veuillez saisir un caractere : ");    scanf(" %c",&ch);

    printf("caractere saisi : %c\t maj toupper= %c\n", ch,toupper(ch));
    printf("caractere saisi : %c\t min tolower= %c\n", ch, tolower(ch));
    putchar('\n');
```

}

LA MOYENNE DE DEUX NOMBRES ENTRÉS AU CLAVIER

Exercice :

écrire un programme qui permet de calculer la moyenne de deux nombres réels entrés au clavier et l'afficher :

LA MOYENNE DE DEUX NOMBRES ENTRÉS AU CLAVIER

Exercice : écrire un programme qui permet de calculer la moyenne de deux nombres réels entrés au clavier et l'afficher :

```
#include <stdio.h>
void main()    // Point d'entré du programme
{
    /*Début du programme
    */
    float x, y;
    printf("\t\t\tCalcul de la moyenne\n\n");           /*Affiche le titre*/
    printf("Entrez le premier nombre : ");
    scanf("%f", &x);                                     /*Lecture du premier nombre*/
    printf("Entrez le deuxieme nombre : ");
    scanf("%f", &y);                                     /*Lecture du deuxième nombre*/
    printf("La valeur moyenne de %.2f et de %.2f est %.2f\n",x, y,(x+y)/2);
} // Fin du programme
```

