

UNIVERSITÉ ABOU BAKR BEL-KAID–TLEMCEN
FACULÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE
Première année licence



Algorithmique et structure de données 1

Chapitre 3: Les structures conditionnelles

Mme HABRI née BENMAHDI Meryem Bochra

Année universitaire: 2024-2025

Plan du cours

1. **Algorithme simple séquentiel**
2. **La prise de décision lors de la résolution d'un problème**
3. **L'importance des structures conditionnelles lors de la résolution d'un problème**
4. **Les opérateurs de comparaison**
5. **Structure conditionnelle simple**
6. **Structure conditionnelle double**
7. **Les opérateurs logiques**
8. **Imbrication de structures conditionnelles**
9. **Structure conditionnelle de choix multiples**
10. **L'opérateur conditionnel ternaire**
11. **Expression vraie et fausse**

Algorithme simple séquentiel

- C'est le type d'algorithme est le plus simple à apprendre pour débiter en programmation.
- Un **algorithme séquentiel simple** est un algorithme dont les instructions s'exécutent **les unes après les autres**, dans l'ordre où elles sont écrites, sans conditions ni répétitions.

La prise de décision lors de la résolution d'un problème

- **En programmation, résoudre un problème ne consiste pas seulement à exécuter des instructions séquentiellement.**
- **Très souvent, il faut choisir une action parmi plusieurs possibilités. C'est à ce moment-là qu'intervient la prise de décision.**
- **Les structures conditionnelles sont des outils qui permettent au programme de prendre une décision automatiquement en fonction de certaines conditions.**
- **Par exemple, si un étudiant a une moyenne supérieure ou égale à 10, le programme peut afficher “Admis”, sinon il affichera “Ajourné”**

L'importance des structures conditionnelles lors de la résolution d'un problème

- **Les structures conditionnelles représentent un outil essentiel de la prise de décision en programmation.**
- **Elles transforment un simple enchaînement d'instructions en un raisonnement logique capable de résoudre des problèmes réels.**
- **L'importance d'utiliser les structures conditionnelles se résume par:**
 - Adapter la solution à la situation
 - Rendre le programme intelligent
 - Gagner en précision et en efficacité
 - Se rapprocher de la logique humaine

Les opérateurs de comparaison

Les **opérateurs de comparaison** (appelés aussi *opérateurs relationnels*) servent à **comparer deux valeurs** dans un algorithme ou un programme.

Symbole	Signification
==	Est égale á
>	Est supérieure á
<	Est inférieure á
>=	Est supérieure ou égale á
<=	Est inférieure ou égale á
!=	Est différent de

Structure conditionnelle simple

- La **structure conditionnelle simple** permet d'exécuter une série d'instructions si jamais une condition **est vraie**.
- La syntaxe:

➤ **En langage algorithmique**

Si (condition) **Alors**

// instructions à exécuter si la condition est vraie

Finsi

➤ **En langage C**

if (condition) {

// instructions à exécuter si la condition est vraie

}

Algorithme d'une structure conditionnelle simple

Exemple: écrire un algorithme qui affiche la valeur absolue d'un entier entré par l'utilisateur, trouver son organigramme et le traduire en programme

Algorithme de la valeur absolue

var nombre: entier;

Début

Lire(nombre)

Si (nombre<0) alors

Nombre ← nombre*(-1);

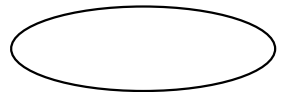
Finsi

Ecrire ('la valeur absolue de ce nombre est ', nombre);

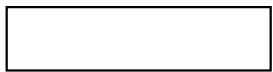
Fin

L'organigramme

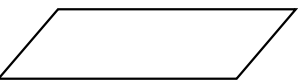
L'organigramme est une représentation graphique de l'enchaînement d'une suite d'actions. Un organigramme utilise des symboles graphiques.



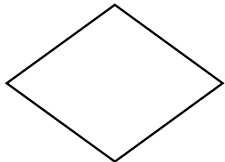
Le début et la fin d'un traitement



Les opérations de traitement



Une entrée ou une sortie

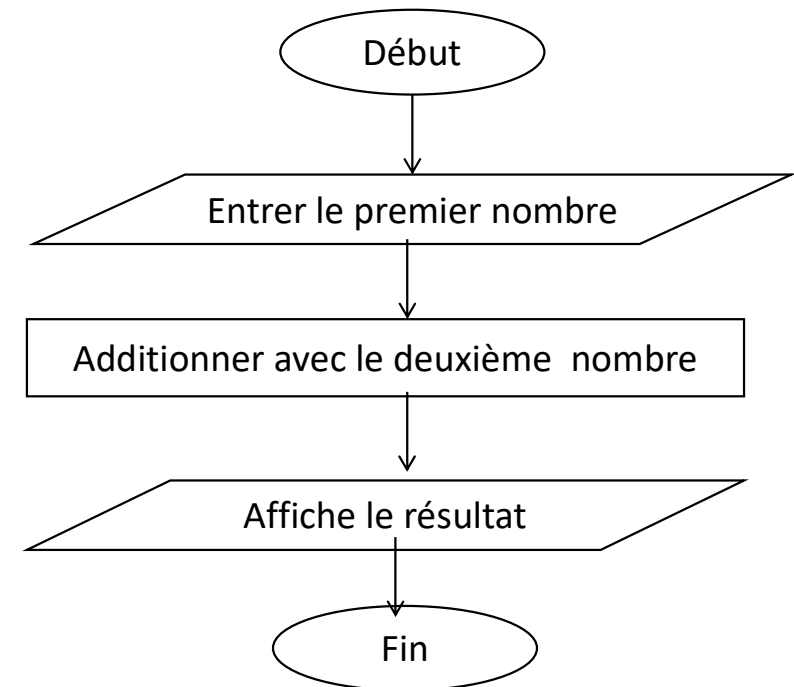


Test (condition)

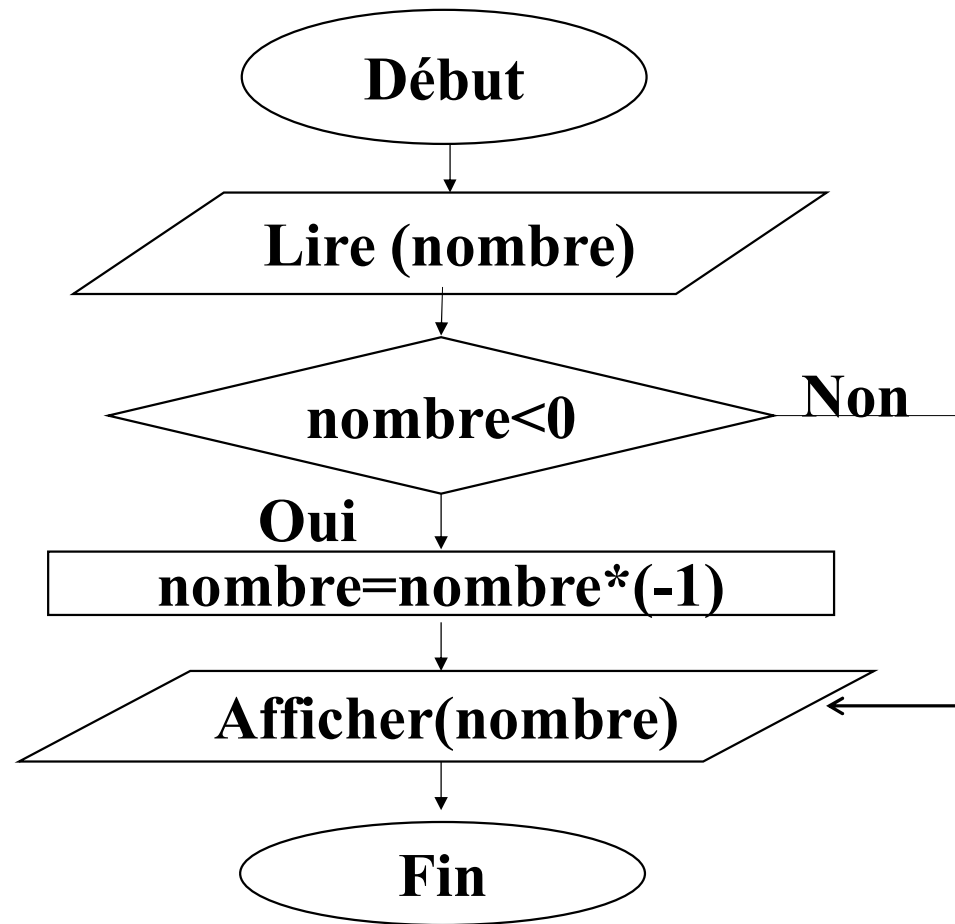


Sens des actions

**Exemple de
l'addition
avec la
calculatrice**



Organigramme d'une structure conditionnelle simple



Programme d'une structure conditionnelle simple

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int nombre;
    printf("entrer un nombre ");
    scanf("%d",&nombre);
    if (nombre<0){
        nombre=nombre*(-1);
    }
    printf("la valeur absolue de ce nombre est %d ",nombre);

    return 0;
}
```

Exercice

Ecrire un algorithme qui permet de faire la division de deux réels entrés par l'utilisateur, trouver l'organigramme ensuite le traduire en programme C. afin d'éviter la division par zéro, vérifier le dénominateur s'il est nul et si c'est le cas il prendra 1 comme valeur par défaut.

Solution: algorithme

Algorithme de la division

var a,b,d: réel;

Début

Afficher('donnez le numérateur');

Lire(a);

Afficher('donnez le dénominateur');

Lire(b);

Si (b=0) alors

$b \leftarrow 1$;

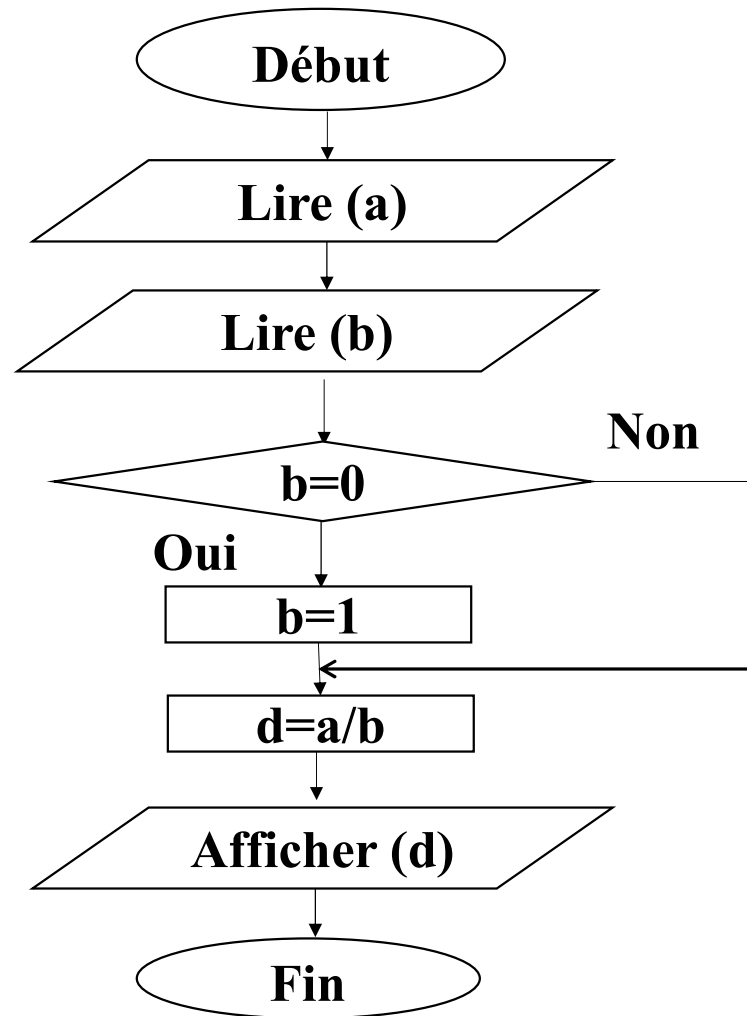
Finsi

$d \leftarrow a/b$;

Ecrire ('le réel recherché est de ', d);

Fin

Solution: organigramme



Solution : programme C

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    double a,b,d;
    printf("entrer le numérateur /n ");
    scanf("%lf",&a);
    printf("entrer le dénominateur /n");
    scanf("%lf",&b);
    if (b==0)
    {b=1;}
    d=a/b;
    printf( "le réel recherché est de %lf",d);
    return 0;
}
```

Structure conditionnelle double

- La **structure conditionnelle double** est une instruction de contrôle qui permet de **choisir entre deux actions** selon qu'une condition est vraie ou fausse.
- La syntaxe:

➤ En langage algorithmique

Si (condition) **Alors**

/* instructions à exécuter si la condition est vraie*/

Sinon

/* instructions à exécuter si la condition est fausse*/

Finsi

➤ En langage C

```
if (condition) {  
    /* instructions à exécuter si la  
    condition est vraie */  
}
```

```
else  
{  
    /* instructions à exécuter si la  
    condition est fausse */  
}
```

Structure conditionnelle double

Exemple: écrire un algorithme qui prend en entrée la moyenne annuelle d'un étudiant et qui affiche "Admis" si la moyenne supérieure ou égale à 10, sinon il affichera "Ajourné". Trouver l'organigramme et le programme C.

Solution: algorithme

Algorithme de la moyenne

var moyenne: réel;

Début

Afficher('entrez votre moyenne annuelle');

Lire(moyenne);

Si (moyenne $\geq$ 10) alors

Afficher('Admis');

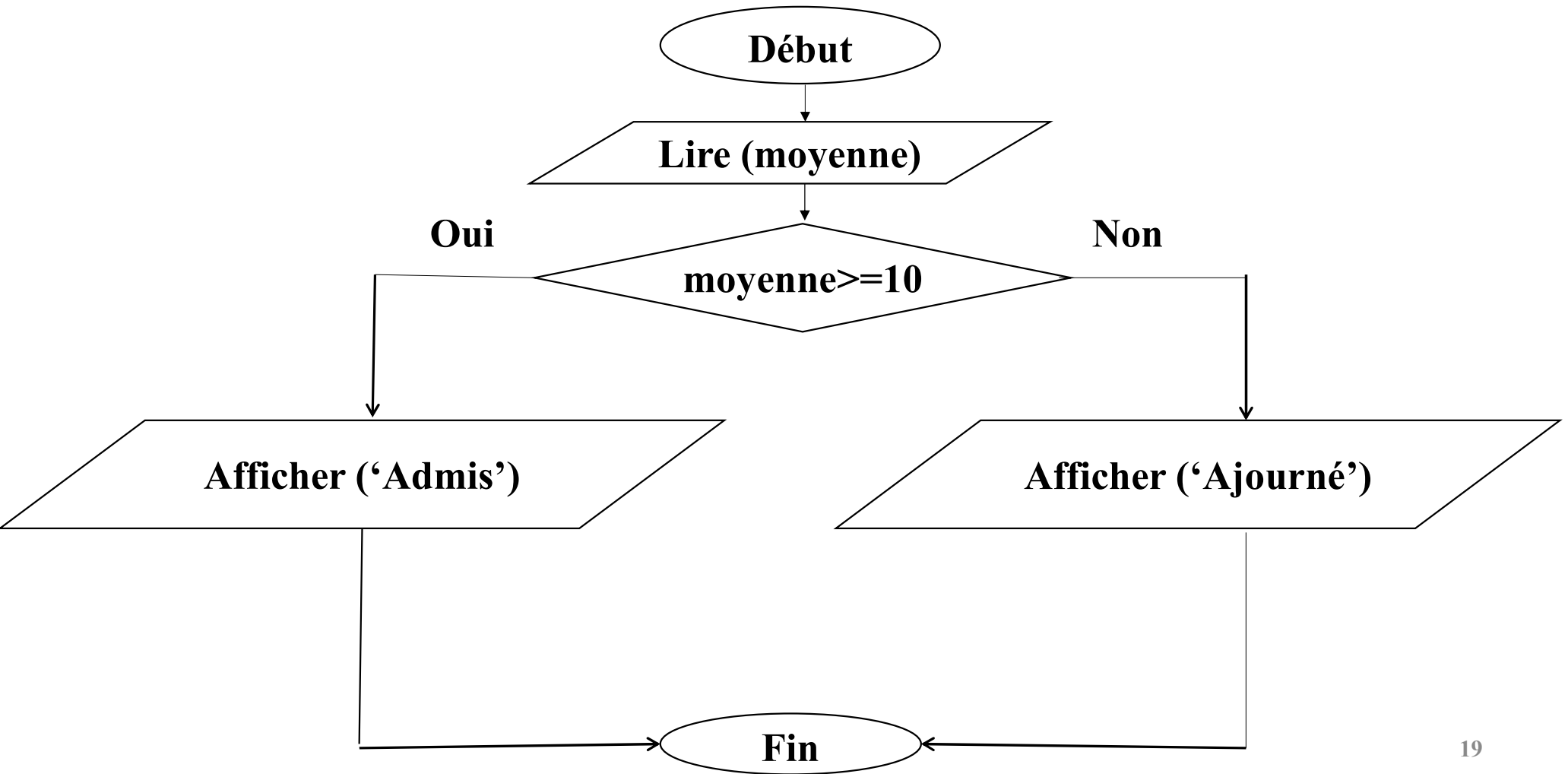
Sinon

Afficher('Ajourné');

Finsi

Fin

Solution: organigramme



Solution : programme C

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    double moyenne;
    printf("entrez votre moyenne annuelle \n ");
    scanf("%lf",&moyenne);
    if (moyenne>=10)
        /* c'est possible d'enlever
           les accolades { } car nous avons une seule instruction*/
        printf("Admis");
    else
        printf("Ajourne ");

    return 0;
}
```

Exercice

Écrire un programme en langage C qui :

- 1. Demande à l'utilisateur de saisir deux entiers (valeur1 et valeur2).**
- 2. Compare les deux entiers saisis.**
- 3. Affiche :**
 - Si la première valeur est inférieure à la deuxième, alors afficher**
 $\text{valeur2} > \text{valeur1}$
 - Sinon, afficher :**
 $\text{valeur1} \geq \text{valeur2}$

Solution

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int valeur1;
    int valeur2;
    printf ("Entrez une 1ere valeur : ");
    scanf ("%d",&valeur1);
    printf ("Entrez 2eme valeur : ");
    scanf ("%d",&valeur2);
    if (valeur1<valeur2)
        printf("%d > %d\n",valeur2, valeur1);
    else
        printf("%d >= %d\n",valeur1, valeur2);
    return 0;
}
```

Les opérateurs logiques

- Les **opérateurs logiques** servent à **combiner plusieurs conditions** dans une structure conditionnelle ou une boucle.
- Ils donnent comme résultat **Vrai** ou **Faux**.

Symbole	Signification
&&	Et
	Ou
!	Non

Exemple

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i1=1;
    int i2=0;
    printf("i1 || i2 = %d",i1||i2);
    printf("i1 && i2 = %d",i1&& i2);
    printf("contraire(1)=%d",!(1));
    return 0;
}
```

Imbrication de structure conditionnelle

- L'imbrication de structures conditionnelles signifie qu'on place une condition à l'intérieur d'une autre condition.
- Cela permet de prendre des décisions plus complexes, en testant plusieurs cas.
- Exemple:

```
#include <stdio.h>
int main() {
    int n;
    printf("Donner un nombre : ");
    scanf("%d", &n);
    if (n >= 0) {
        if (n == 0) {
            printf("Le nombre est nul\n");
        } else {
            printf("Le nombre est positif\n");
        }
    } else {
        printf("Le nombre est négatif\n");
    }
    return 0;
}
```

Structure conditionnelle de choix multiples

- Une structure de contrôle conditionnelle est dite à choix lorsque le traitement dépend de la valeur que prendra un sélecteur.
- Ce sélecteur est de type entier, caractère ou booléen.
- En algorithmique, on l'appelle souvent "structure selon" ou "switch – case" en langage C

Syntaxe de la structure conditionnelle de choix multiples

➤ *En pseudo code*

Selon selecteur faire

cas Valeur 1:

//Liste d'instructions;

cas Valeur 2:

//Liste d'instructions;

cas Valeur ...:

//Liste d'instructions;

sinon :

//Liste d'instructions par défaut;

FinSelon

➤ *En langage C*

switch (selecteur)

{

case Valeur1 :

//Liste d'instructions;

break;

case Valeur2 :

//Liste d'instructions;

break;

case Valeurs... :

//Liste d'instructions;

break;

default:

//Liste d'instructions par défaut;}

N'oubliez pas d'insérer des instructions break entre chaque test

Exercice d'application

Écrire un algorithme, trouver l'organigramme et le programme C qui :

- 1. Demande à l'utilisateur de saisir un numéro représentant un jour de la semaine (un entier entre 1 et 7).**
- 2. Affiche le nom du jour correspondant :**

1 → Dimanche

2 → Lundi

3 → Mardi

4 → Mercredi

5 → Jeudi

6 → Vendredi

7 → Samedi

Si l'utilisateur entre un nombre en dehors de l'intervalle [1,7], le programme doit afficher un message d'erreur : "Numéro de jour invalide"

Solution : algorithme

Algorithme du jour de la semaine

Début

Écrire ("Donnez un numéro entre 1 et 7 : ")

Lire (jour)

Selon jour faire

cas 1 : Écrire ("Dimanche")

cas 2 : Écrire ("Lundi")

cas 3 : Écrire ("Mardi")

cas 4 : Écrire ("Mercredi")

cas 5 : Écrire ("Jeudi")

cas 6 : Écrire ("Vendredi")

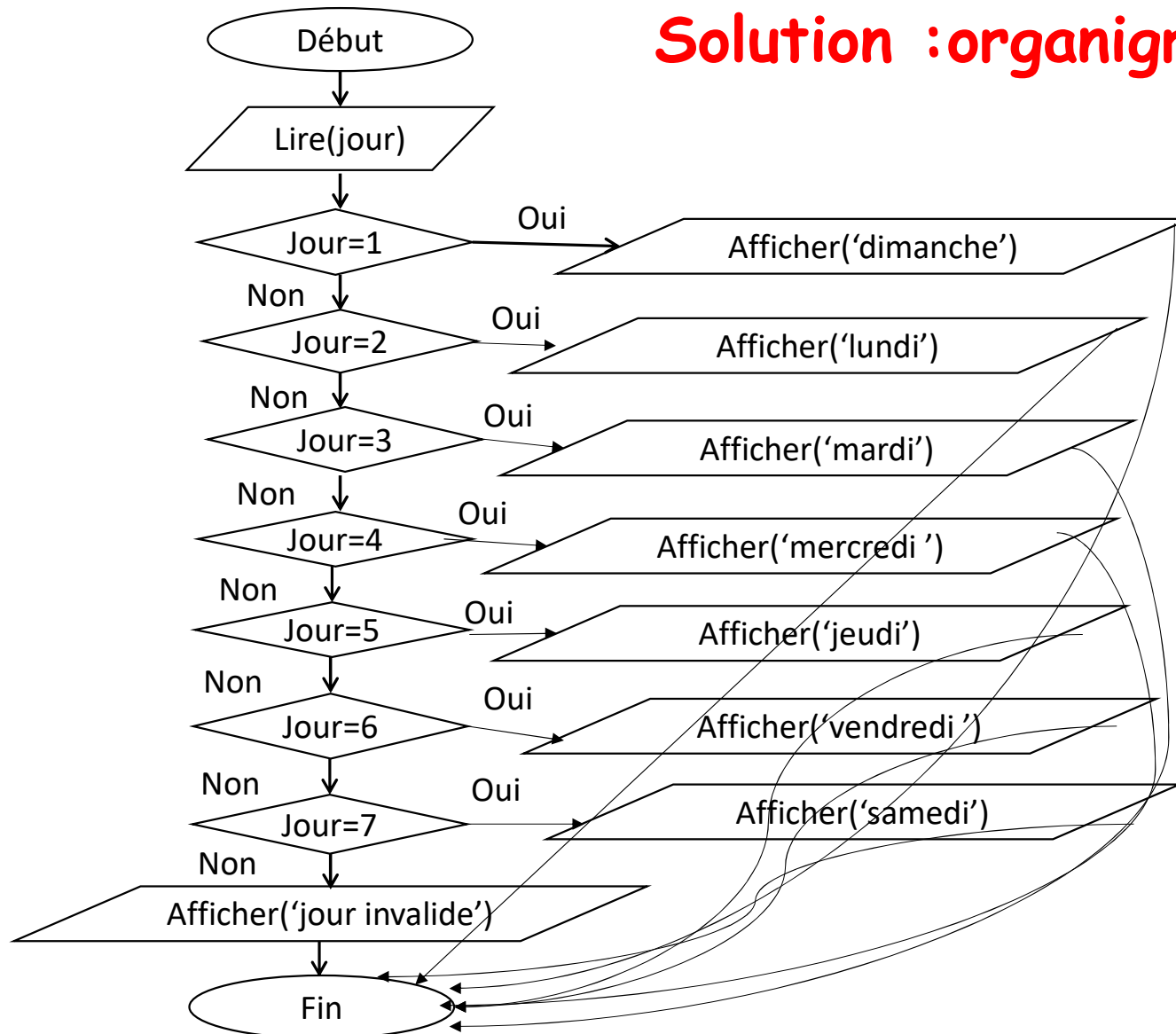
cas 7 : Écrire ("Samedi")

sinon : Écrire ("Numéro de jour invalide !")

FinSelon

Fin

Solution : organigramme



Solution :programme C

```
#include <stdio.h>
int main() {
    int jour;
    printf("Donnez un numéro entre
1 et 7 : ");
    scanf("%d", &jour);
    switch(jour) {
        case 1:
            printf("Dimanche\n");
            break;
        case 2:
            printf("Lundi\n");
            break;
        case 3:
            printf("Mardi\n");
            break;
```

```
        case 4:
            printf("Mercredi\n");
            break;
        case 5:
            printf("Jeudi\n");
            break;
        case 6:
            printf("Vendredi\n");
            break;
        case 7:
            printf("Samedi\n");
            break;
        default:
            printf("Numéro de jour invalide !\n");
    }
    return 0;
}
```

L'opérateur conditionnel ternaire

C'est une **forme simplifiée du if...else**, pratique quand on a seulement deux choix.

Syntaxe:

➤ En pseudo code
(condition) ? valeur_si_vrai : valeur_si_faux

➤ En langage C
(condition) ? valeur_si_vrai : valeur_si_faux;

Remarques :

- la condition doit être entre des parenthèses
- Lorsque la condition est vraie, l'instruction de gauche est exécutée
- Lorsque la condition est fausse, l'instruction de droite est exécutée
- En plus d'être exécutée, la structure ? : renvoie la valeur résultante

Exemple

Exemple en pseudo code:

```
(moyenne >=10) ? afficher ("Admis") : afficher("Ajourne");  
admis ← ((moyenne >=10) ? 1 : 0);
```

Exemple en langage C:

```
(moyenne >=10) ? printf ("Admis") : printf ("Ajourne");  
admis = ((moyenne >=10) ? 1 : 0);
```

Exercice d'application

Ecrire un programme c qui permet de lire deux réels dont les valeurs sont différentes, et qui affiche le minimum entre ces deux nombres.(utiliser l'opérateur ternaire et trois variables)

```
int main()
{ double a,b,min;
  printf("entrer deux nombres differents\n");
  scanf("%lf",&a);
  scanf("%lf",&b);
  if(a==b)
    printf("les deux reels ont la meme valeur");
  else{
    min=(a<b)?a:b;
    printf("le minimun entre %lf et %lf est de %lf",a,b,min);}
  return 0;
}
```

Expression vraie et fausse

- La valeur Vrai peut être assimilée à la valeur numérique 1 ou à toute valeur non nulle.
- La valeur Faux peut être assimilée à la valeur numérique 0.
- Ne pas oublier les parenthèses lorsqu'il y a un if.

Exemple 1 :

```
if (0)
{ printf("C'est faux"); }
else
{ printf("C'est vrai"); }
```

Si (0) alors	Afficher('C'est faux');
Sinon	Afficher('C'est vrai');
Finsi	
Fin	

Exercice d'application

Écrire un pseudocode qui demande à l'utilisateur de **saisir une heure (entière)** comprise entre **0 et 23**, puis affiche à quel **moment de la journée** cette heure correspond selon les règles suivantes :

- de **6 à 11** → afficher : C'est le matin
- de **12 à 16** → afficher : C'est l'après-midi
- de **17 à 19** → afficher : C'est le soir
- de **20 à 23** et de **0 à 5** → afficher : C'est la nuit

Solution en pseudo code

Algorithme d'affichage moment de la journée

Var heure:entier;

Début

Lire(heure)

Si heure $\geq$ 6 et heure $\leq$ 11 alors

Écrire ("C'est le matin")

Sinon si heure $\geq$ 12 et heure $\leq$ 16 alors

Écrire ("C'est l'après-midi")

Sinon si heure $\geq$ 17 et heure $\leq$ 19 alors

Écrire ("C'est le soir")

Sinon si (heure $\geq$ 20 et heure $\leq$ 23) ou (heure $\geq$ 0 et heure $\leq$ 5) alors

Écrire ("C'est la nuit« ")

Sinon

Écrire ("Heure invalide ! ")

FinSi

Fin

Solution en programme C

```
int main()
{if (heure >= 6 && heure <= 11)
    printf("C'est le matin\n");
else if (heure >= 12 && heure <= 16)
    printf("C'est l'après-midi\n");
else if (heure >= 17 && heure <= 19)
    printf("C'est le soir\n");
else if ((heure >= 20 && heure <= 23) || (heure >= 0 && heure <= 5))
    printf("C'est la nuit\n");
else
    printf("Heure invalide !\n");
return 0;}
```