

Chapitre 2: Les structures alternatives

Algorithmique & Pascal

adrien.basse@uadb.edu.sn

**Les structures alternatives
permettent de choisir les
actions à exécuter suivant la
réalisation d'une condition.**

Nous disposons d'informations sur le nom et l'âge d'une personne et nous désirons afficher ces informations uniquement si l'âge est supérieur à 18.

- Si l'âge est supérieur à 18 alors nous effectuons un affichage du nom et de l'âge.
- Dans le cas contraire, aucune action n'est réalisée.

Nous disposons de 3 valeurs entières a, b et c et nous voulons afficher les valeurs entières positives parmi ces valeurs.

Pour chaque valeur, nous réalisons un test pour savoir si elle est positive.

- Si tel est le cas la valeur est affichée.**
- Dans le cas contraire aucune action n'est réalisée.**

Nous disposons de 3 valeurs entières (a, b et c) et nous voulons calculer et afficher le nombre de valeurs entières négatives parmi ces valeurs.

Pour chaque valeur, nous réalisons un test pour savoir si elle est négative.

- Si tel est le cas, cette valeur est comptabilisée parmi les valeurs négatives.**
- Dans le cas contraire, aucune action n'est réalisée.**

Nous disposons de 3 valeurs entières (a, b et c) et nous voulons calculer et afficher la somme des valeurs entières positives parmi celles ci.

Pour chaque valeur, nous réalisons un test pour savoir si elle est positive.

- **Si tel est le cas, cette valeur est comptabilisée dans la somme.**
- **Dans le cas contraire, aucune action n'est réalisée.**

Et pour les valeurs nulles? Et pour le produit des valeurs?

Nous disposons de deux valeurs entières et nous désirons afficher le plus grand des deux.

Nous réalisons un test pour savoir si la première valeur est plus grande que la deuxième.

- Si tel est le cas alors la première valeur est affichée.**
- Dans le cas contraire, la deuxième valeur est affichée.**

Et pour les afficher dans l'ordre croissant?

Nous disposons de 3 valeurs entières (a, b et c) et nous voulons calculer et afficher la somme des valeurs entières positives, la somme des valeurs négatives, la somme de toutes les valeurs.

Pour chaque valeur, nous réalisons un test pour savoir si elle est positive.

- Si tel est le cas, cette valeur est comptabilisée dans la somme des valeurs positives.
- Dans le cas contraire, la valeur correspondante est comptabilisée dans la somme des valeurs négatives.
- Toutes les valeurs, sans aucun test, sont comptabilisées dans la somme totale.

Nous disposons d'informations sur le nom et l'âge d'une personne et nous désirons afficher le nom de la personne avec la mention « majeur » ou « mineur » selon que son âge est supérieur ou inférieur à 18.

Nous réalisons un test pour savoir si l'âge est supérieure à 18.

- **Si tel est le cas alors le nom de la personne est affiché avec la mention « Majeur ».**
- **Dans le cas contraire, le nom est affiché avec la mention « mineur ».**

Nous disposons de deux valeurs entières et nous désirons afficher l'un des quatre messages suivants : « a et b sont positifs », « a et b sont négatifs », « a est positif et b est négatif », « a est négatif et b est positif ».

Contrairement aux cas précédents, le test ne porte pas sur une seule valeur mais sur plusieurs (2) avec un ET de liaison.

**Nous disposons de deux valeurs entières
et nous désirons afficher si cela est vrai :
« a et b sont de signes contraires ».**

- Une première solution consiste à faire un test complexe comme dans le cas précédent sur le signe de a et de b .
- Une deuxième solution moins intuitive mais plus simple consiste à utiliser le fait que lorsque deux nombres sont de signes contraires leur produit est négatif.

Nous disposons de 3 valeurs entières (a, b et c) et nous voulons calculer et afficher les valeurs entières comprises entre 0 (inclus) et 20 (non inclus) parmi ces valeurs.

Pour chaque valeur, nous réalisons un test complexe pour savoir si elle est supérieure ou égal à zéro et strictement inférieur à 20.

- Si tel est le cas, la valeur correspondante est affichée.
- Dans le cas contraire, aucune action n'est réalisée.

Mise en œuvre

Pseudo code, Algorigramme, Pascal

A Venn diagram consisting of two concentric circles. The outer circle is orange and contains the text 'Choix simple'. The inner circle is purple and contains the text 'Sélection multiple'. The inner circle is entirely contained within the outer circle, indicating that 'Sélection multiple' is a subset of 'Choix simple'.

Choix
simple

Sélection
multiple

Pseudo code

Choix simple sans alternative

Si condition alors

Bloc d'actions

finsi

**Nous disposons de trois valeurs
entière a, b, c et nous voulons
afficher les valeurs entières positives
parmi ces valeurs.**

A compléter ...

```
Si  $a > 0$  alors  
    écrire (a, 'est positif')  
finsi
```

**Nous disposons de deux valeurs
entière a , b et nous voulons afficher,
si c'est possible, le résultat de la
division de a par b .**

A compléter ...

```
Si b <> 0 alors  
  c ← a/b  
  écrire ('Resultat est ',c)  
finsi
```

Pseudo code

Choix simple avec alternative

Si condition alors

Bloc d'actions

Sinon

Bloc d'actions

finsi

Nous disposons de trois valeurs entières a , b , c et nous voulons les afficher en précisant leurs signes (positif ou négatif).

NB: Les valeurs nulles sont considérées comme positives.

A compléter ...

Si $a \geq 0$ alors

 écrire (a, ' est positif')

Sinon

 écrire (a, ' est négatif')

finsi

Nous disposons de deux valeurs entière a , b et nous voulons afficher le résultat de la division de a par b ou un message d'erreur si la division est impossible.

Si $b \neq 0$ alors

$c \leftarrow a/b$

écrire ('Resultat est',c)

Sinon

écrire ('Impossible')

finsi

Pseudo code

Choix simple imbriqué

Si condition alors

Si condition alors

Bloc d'actions

finSi

finSi

...

Sinon

Si condition alors

Bloc d'actions

finSi

finSi

Nous disposons de trois valeurs entière a, b, c et nous voulons les afficher en précisant leurs signes (positif, négatif ou nul).

A compléter ...

Si $a > 0$ **alors**

ecrire (a , 'est positif')

Sinon

Si $a < 0$ **alors**

ecrire (a , 'est négatif')

Sinon

ecrire (a , 'est nul')

finsi

finsi

Indentez votre code !

Si $a > 0$ **alors**

ecrire (a , 'est positif')

Sinon

Si $a < 0$ **alors**

ecrire (a , 'est négatif')

Sinon

ecrire (a , 'est nul')

finsi

finsi

Algorithme

Compteur d'éléments

Etape 1: Initialisation

compteur $\leftarrow 0$;

Etape 2: Incrémentation

compteur \leftarrow compteur + 1

L'initialisation est réalisée une seule fois avant de commencer le décompte.

L'incrémentation est réalisée pour chaque élément du décompte.

**Nous disposons de trois valeurs
entière a, b, c et nous voulons afficher
le nombre de valeurs positives parmi
celles ci.**

A compléter ...

```
compteur ← 0;  
Si a > 0 alors  
    compteur ← compteur + 1;  
finsi  
Si b > 0 alors  
    compteur ← compteur + 1;  
finsi  
Si c > 0 alors  
    compteur ← compteur + 1;  
finsi  
ecrire (compteur, 'valeurs positives');
```

Algorithme

Somme d'éléments

Etape 1: Initialisation

somme \leftarrow 0;

Etape 2: Incrémentation

somme \leftarrow somme + element

L'initialisation est réalisée une seule fois avant de débiter la sommation.

L'incrémentation est réalisée pour chaque élément à ajouter à la somme.

**Nous disposons de trois valeurs
entière a, b, c et nous voulons afficher
la somme des valeurs négatives parmi
celles ci.**

A compléter ...

somme \leftarrow 0 ;

Si a < 0 alors

somme \leftarrow somme + a;

finsi

Si b < 0 alors

somme \leftarrow somme + b;

finsi

Si c < 0 alors

somme \leftarrow somme + c;

finsi

ecrire (Somme des valeurs negatives est:', somme);

Algorithme

Produit d'éléments

Etape 1: Initialisation

produit \leftarrow 1;

Etape 2: multiplication

produit \leftarrow produit * element;

L'initialisation est réalisée une seule fois avant de débiter le produit.

La multiplication est réalisé pour chaque élément à multiplier au produit.

**Nous disposons de trois valeurs
entière a, b, c et nous voulons afficher
le produits des valeurs non nulles
parmi celles ci.**

A compléter ...

produit \leftarrow 1;

Si $a \neq 0$ alors

****produit \leftarrow produit * a;****

finsi

Si $b \neq 0$ alors

****produit \leftarrow produit * b;****

finsi

Si $c \neq 0$ alors

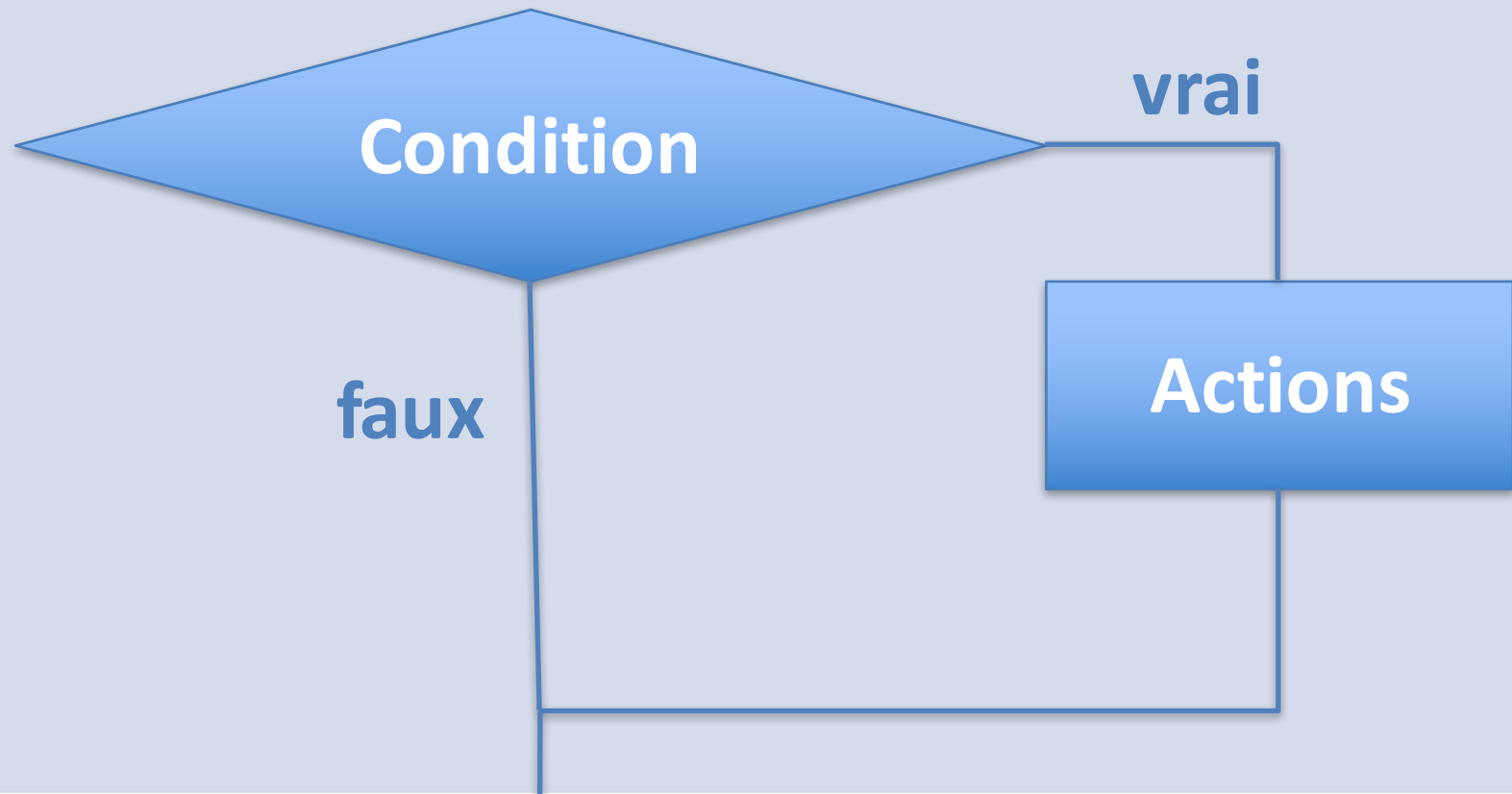
****produit \leftarrow produit * c;****

finsi

ecrire (Produit des valeurs non nulles :', produit);

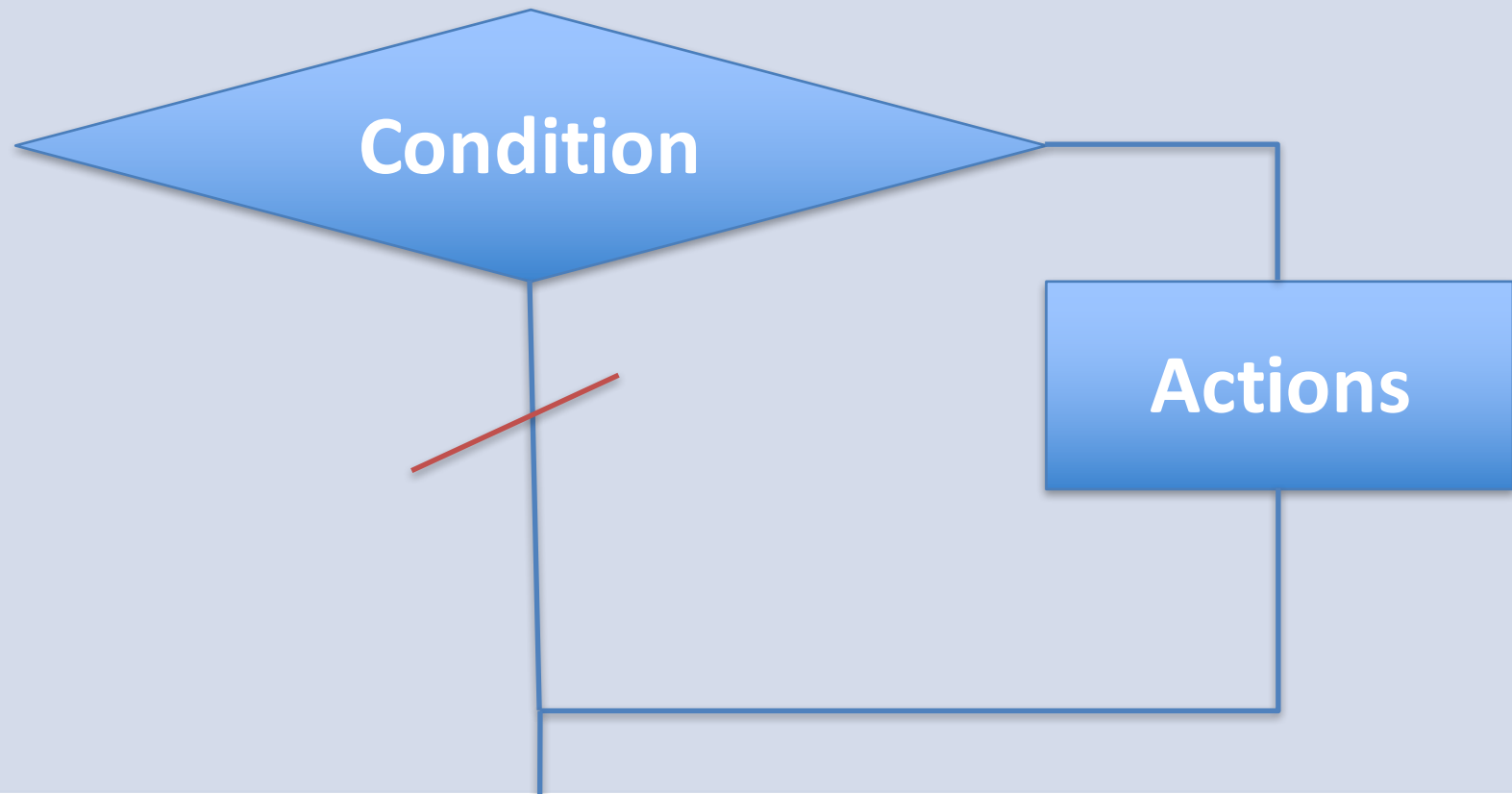
Algorithme

Choix simple sans alternative



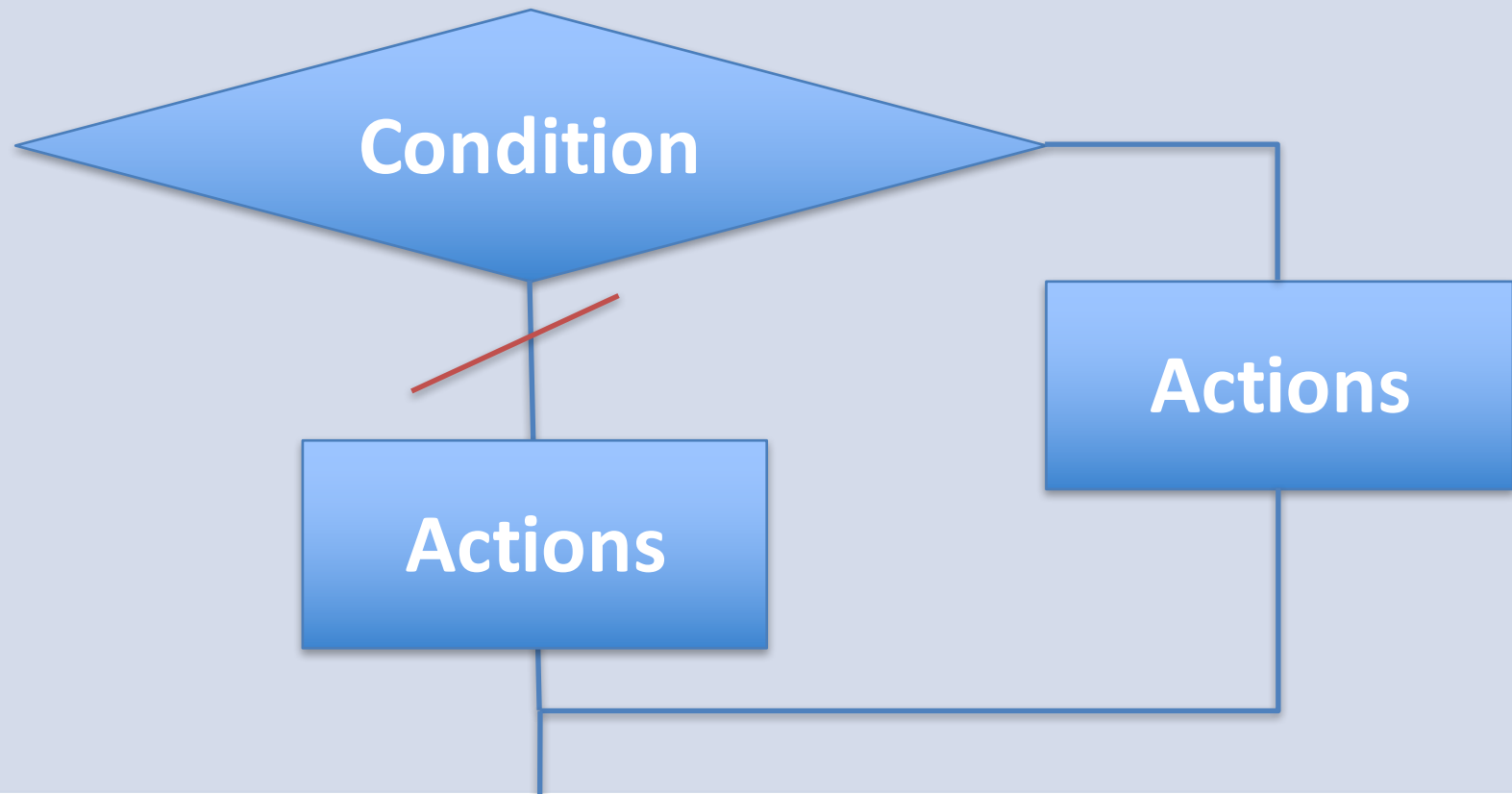
Algorithme

Choix simple sans alternative



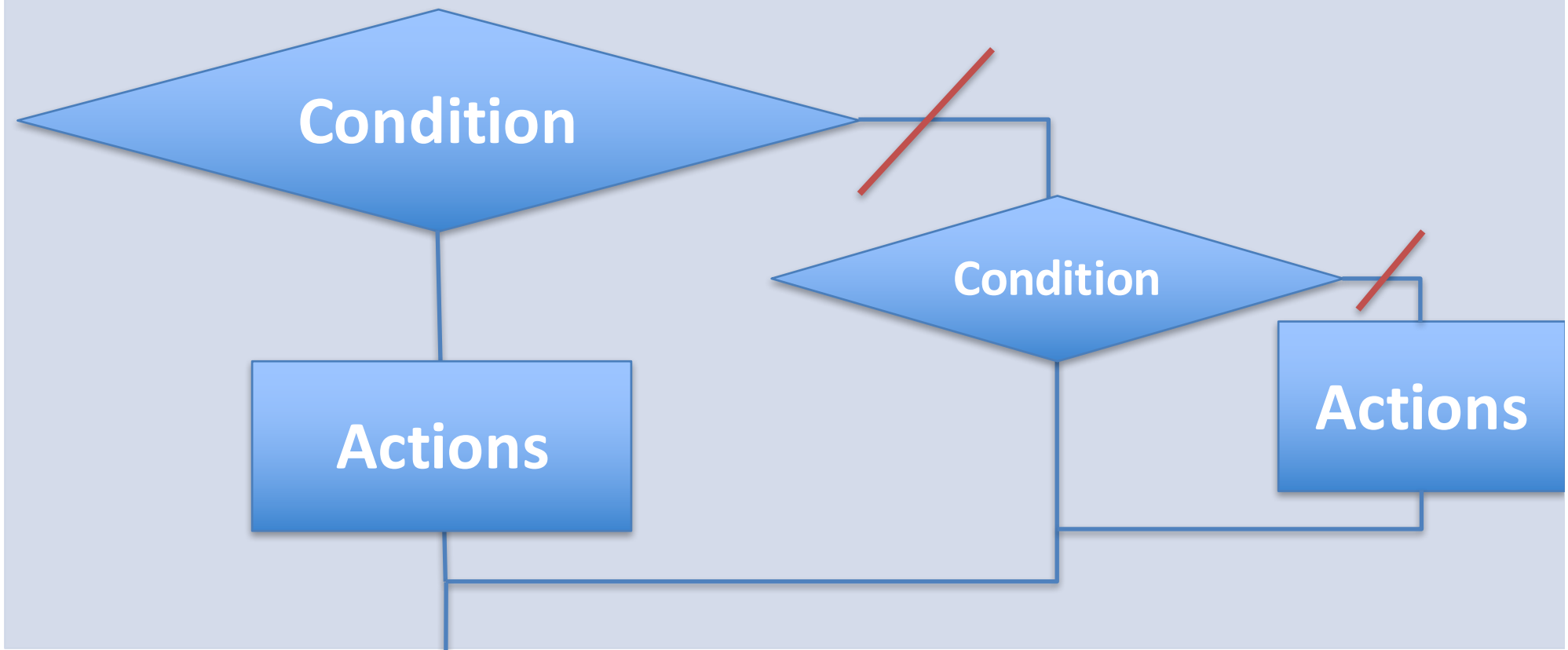
Algorithme

Choix simple avec alternative



Algorithme

Choix simple imbriqué



Proposer un algorithme...

Nous disposons de trois valeurs entière a, b, c et nous voulons afficher les valeurs entières positives parmi ces valeurs.

Proposer un algorithme...

Nous disposons de deux valeurs entières a , b et nous voulons afficher, si c'est possible, le résultat de la division de a par b .

Proposer un algorithme...

Nous disposons de trois valeurs entière a, b, c et nous voulons les afficher en précisant leurs signes (positif ou négatif).

NB: Les valeurs nulles sont considérées comme positives.

Proposer un algorithme...

Nous disposons de deux valeurs entière a , b et nous voulons afficher le résultat de la division de a par b ou un message d'erreur si la division est impossible.

Proposer un algorithme...

Nous disposons de trois valeurs entières a , b , c et nous voulons les afficher en précisant leurs signes (positif, négatif ou nul).

Proposer un algorithme...

Nous disposons de trois valeurs entières a, b, c et nous voulons afficher le nombre de valeurs positives parmi celles ci.

Proposer un algorithme...

Nous disposons de trois valeurs entières a, b, c et nous voulons afficher la somme des valeurs négatives parmi celles ci.

Proposer un algorithme...

Nous disposons de trois valeurs entières a, b, c et nous voulons afficher le produits **des valeurs non nulles parmi celles ci.**

Pascal

Choix simple sans alternative

```
if condition then  
    instruction;
```



```
if a > 0 then  
  write (a, ' est positif');
```

Attention: L'instruction if
contient par défaut une seule
Instruction.

Résultat ???

```
a := - 10;  
if a > 0 then  
    write (a, ' est positif');  
write ('a vaut ',a);
```

Résultat ???

```
a := - 10;  
if a > 0 then  
    write (a, ' est positif');  
write ('a vaut ',a);
```

Résultat ???

```
a := 10;  
b:=0;  
if b <> 0 then  
    c := a/b;  
write ('Résultat est ',c);
```

Un **bloc d'instructions** permet de disposer de plusieurs instructions là où par défaut une seule instruction est possible.

En pascal, un bloc est délimité entre

Begin

et

end;

```
if b <> 0 then  
begin  
    c := a/b;  
    write ('Resultat est ',c);  
end;
```


Begin et end est optionnel pour une seule instruction

```
if a > 0 then  
  begin  
    write (a, ' est positif');  
  end;
```

Pascal

Choix simple avec alternative

```
if condition then  
    Instruction  
else  
    Instruction;
```

~~;~~
else

```
if b <> 0 then
begin
  c := a/b;
  write ('Resultat est',c);
end
else
  write ('Impossible') ;
```

Pascal

Choix simple imbriqué

```
if condition then  
    if condition then  
        Instruction;
```

...

else

if condition then

instruction;

Proposer un programme Pascal

Nous disposons de trois valeurs entière a, b, c et nous voulons afficher les valeurs entières positives parmi ces valeurs.

Proposer un programme Pascal

Nous disposons de deux valeurs entière a, b et nous voulons afficher, si c'est possible, le résultat de la division de a par b.

Proposer un programme Pascal

Nous disposons de trois valeurs entière a, b, c et nous voulons les afficher en précisant leurs signes (positif ou négatif).

NB: Les valeurs nulles sont considérées comme positives.

Proposer un programme Pascal

Nous disposons de deux valeurs entière a, b et nous voulons afficher le résultat de la division de a par b ou un message d'erreur si la division est impossible.

Proposer un programme Pascal

Nous disposons de trois valeurs entière a, b, c et nous voulons les afficher en précisant leurs signes (positif, négatif ou nul).

Proposer un programme Pascal

Nous disposons de trois valeurs entières a, b, c et nous voulons afficher le nombre de valeurs positives parmi celles ci.

Proposer un programme Pascal

Nous disposons de trois valeurs entières a, b, c et nous voulons afficher la somme des valeurs négatives parmi celles ci.

Proposer un programme Pascal

Nous disposons de trois valeurs entières a, b, c et nous voulons afficher le produits des valeurs non nulles parmi celles ci.

**La sélection multiple permet de
choisir les actions à exécuter pour
différentes valeurs d'une donnée
entière ou caractère.**

Sélection multiple

Cas donnee parmi

Constante 1: Bloc d'actions

Constante 2: Bloc d'actions

Intervalle: Bloc d'actions

...

Sinon: Bloc d'actions

finCas

Si donnee=constante1

Bloc d'actions

sinon si donnee=constante2

Bloc d'actions

sinon si donnee dans intervalle

Bloc d'actions

...

Intervalle d'entiers

- 4 .. 12

Intervalle de caractères

- 'e' .. 'p'

Sélection multiple

Case variable of

Constante 1: Instructions

Constante 2: Instructions

Intervalle: Instructions

...

else: Instructions

end;

**Réaliser une calculette simple en
Pascal avec la saisie d'une opération:**

entier1 op entier2

et l'affichage du résultat.

op: +, -, *, /

Gérer les cas suivants:

- **Division par zéro,**
- **Opérateur incorrect.**

Quelle est la syntaxe pour mettre un bloc de plusieurs instructions dans un if?

Quelle est la syntaxe pour mettre un bloc de plusieurs instructions dans un si?

Vrai ou faux, tout *if* peut se traduire en *case*?

Le code suivant comporte des erreurs sémantiques et syntaxiques. Lesquelles?

If longueur > largeur

Write ('Correct');

*Perimetre=longueur+largeur*2;*

Else

Write (Incorrecte');

Que fait ce bout de code? Faites des propositions d'amélioration.

```
If rayon > 0 then  
begin  
Write ( 'Correct' );  
end;  
Else  
Write ( 'Incorrecte' );  
Perimetre := 2 * PI * rayon;
```

Proposer un code équivalent avec case.

```
If menu= 0 then  
write ( 'Boisson' );  
Else if menu >0 and menu <5 then  
write ( 'Sandwich' );  
Else if menu=5 then  
write ( 'Plat' );  
Else  
write ( 'Erreur de choix' );
```

Proposer un algorithme qui vérifie si un nombre est strictement positif.

Proposer un algorithme qui vérifie si un nombre n n'est pas strictement positif.

Proposer un algorithme qui vérifie si un nombre est plus grand que 100.

Proposer un algorithme qui vérifie si un nombre est strictement compris entre 1 et 100.

Proposer un algorithme qui vérifie si un nombre est strictement compris entre 1 et 100 sans être égal à 50.

Proposer un algorithme de calcul du signe du produit de deux nombres entiers.

Proposer un algorithme qui vérifie si un nombre est plus grand que trois autres nombres.

Proposer un algorithme qui vérifie si un nombre est plus petit que deux autres nombres tout en restant plus grand que 100.

Proposer un algorithme qui vérifie si un nombre est plus petit que deux autres nombres tout en restant plus grand qu'un troisième.

Proposer un algorithme qui vérifie si un nombre est plus petit que deux autres nombres tout ou plus grand qu'un troisième.

Proposer un algorithme qui vérifie si un nombre est compris entre deux nombres donnés.

Proposer un algorithme qui vérifie si un caractère est une voyelle.

Proposer un algorithme qui vérifie si un caractère est une consonne.

