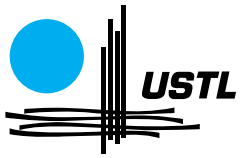


Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

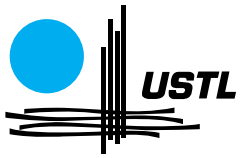
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

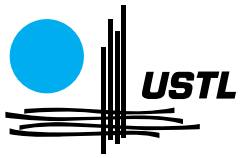
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

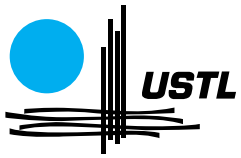
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

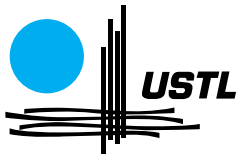
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

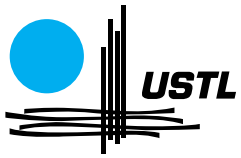
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

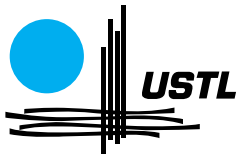
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

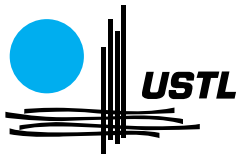
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

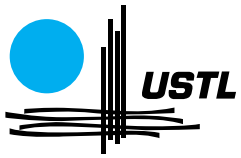
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

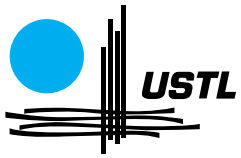
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

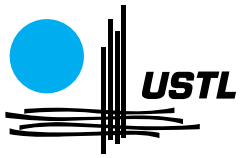
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

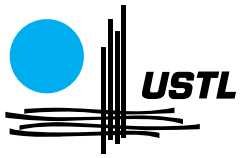
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

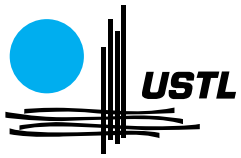
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

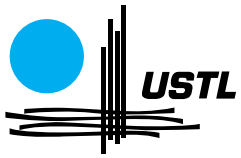
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

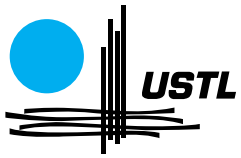
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

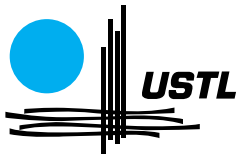
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

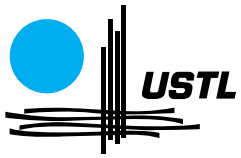
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

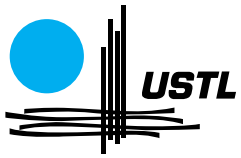
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

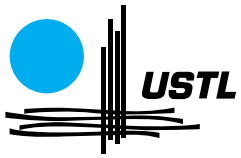
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

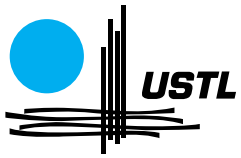
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

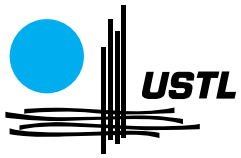
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

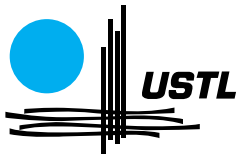
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

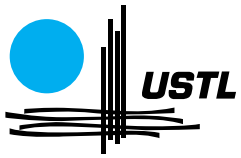
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

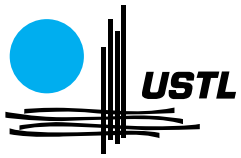
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

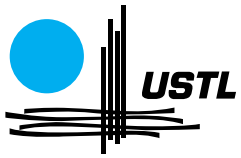
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

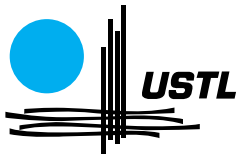
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

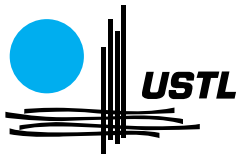
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

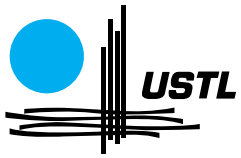
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

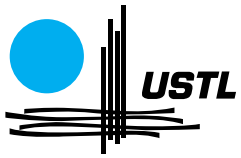
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

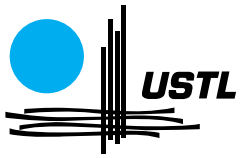
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

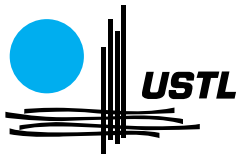
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

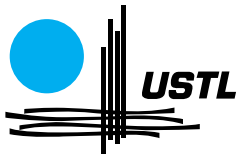
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

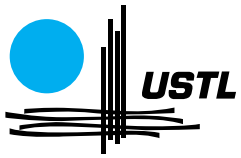
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

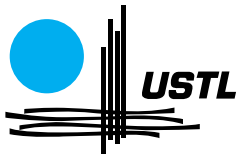
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

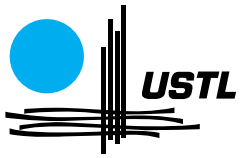
FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet



Maîtrise d'informatique
Module de Projet 1

Examen – Compilateur Pascal

Philippe MARQUET

Janvier 2002

Documents autorisés — Durée : 2 heures

Soit PPx une extension de PP2 (chapitre 6, Variables de type tableau) introduisant la nouvelle structure de contrôle suivante :

```
INST  ::=  INSTS | AFFEC | SI | TANTQUE | ECRIRE | LIRE | ALT
ALT   ::=  alt COND -> INST { ; COND -> INST } end
```

L'instruction `alt` est une suite d'instructions gardées ; chaque instruction gardée est constituée d'une instruction précédée d'une condition, la garde. Si une des gardes est évaluée à vrai, une d'entre elles est arbitrairement sélectionnée et l'instruction correspondante exécutée. Si les gardes sont toutes fausses, l'exécution du programme s'arrête sur une erreur.

Cette instruction est illustrée par l'exemple de la figure 1.

Exercice 1 (Analyse syntaxique)
Donner une procédure d'analyse syntaxique pour l'instruction `alt`.

Exercice 2 (Alternative à `alt`)
Discuter brièvement de la pertinence de cette construction `alt`, en particulier au regard de structures plus traditionnelles telle le `if/elseif/else`.

Exercice 3 (Schéma de génération de P-Code)
Donner le schéma de P-Code à générer pour une instruction `alt`. On pourra étendre le P-Code pour assurer la gestion de la terminaison erronée d'une structure `alt`.

```
program sdico ;
const SIZE = 1023
var inf, sup, val, essai ;
    t[SIZE];
begin
    sup := SIZE ; inf := 0 ;
    read(val) ;
    essai = (sup+inf)/2 ;
    while t[essai] <> val do begin
        alt
            val < t[essai] -> sup := essai-1 ;
            val > t[essai] -> inf := essai+1 ;
        end ;
        essai = (sup+inf)/2 ;
    end ;
    write(essai) ;
end.
```

FIG. 1 – Exemple d'utilisation de la structure `alt`

Id: xam02.tex,v 1.1 2002/01/21 07:32:44 marquet Exp

Exercice 4 (Traduction en P-Code)
Traduire en P-Code le programme de la figure 1.

Exercice 5 (Génération de code)
Étendre la procédure d'analyse syntaxique de l'exercice 1 pour y inclure les instructions de génération de P-Code.

Fin du sujet