

# Table des matières

<i>Aux enseignants</i>	XIII
Pourquoi des études de cas ?	XIII
Caractéristiques des études de cas	XIV
L'utilisation des études de cas	XIX
Remerciements	XXIII
<i>Aux étudiants</i>	XXV
Comment utiliser les études de cas	XXV
Conventions d'écriture	XXVII

## CHAPITRE 1 *Vérifiez ce code!* 1

Chaque jour de plus en plus d'activités sont contrôlées par des codes. Nous avons maintenant des codes numériques pour obtenir de l'argent, pour nous inscrire à un cours, pour avoir accès à un service Minitel, pour activer un système d'alarme, ou pour utiliser une photocopieuse. Le plus souvent la validité de ces codes est déterminée par un ordinateur. Cette étude de cas décrit un programme qui contrôle la validité des codes destinés à une photocopieuse.

PROBLÈME	
Enoncé du problème	1
Préparation	2
DISCUSSION	
Comprendre le problème	2
Une solution n'utilisant que des variables de type entier	4
Amélioration du programme utilisant le type entier	10
Une solution utilisant des variables de type caractère	13
RÉVISION	
Survol des questions d'analyse et de développement	15
Retour sur la programmation	16
Synthèse de <i>Vérifiez ce code!</i>	17
SOURCES PASCAL	
Solution utilisant des variables de type entier	18
Solution améliorée avec des variables de type entier	19
Solution utilisant des variables de type caractère	20

## CHAPITRE 2 *La bannière est dans le SAC*

21

Depuis l'Antiquité les bannières ont proclamé les événements et annoncé les bonnes affaires. Les ordinateurs permettent à tout le monde de créer des bannières personnalisées de tailles et de styles variés. Ce programme imprime des pavés de six lettres dans la taille choisie par l'utilisateur en utilisant les lettres P,A,S,C, et L. Il peut être modifié pour créer des bannières avec des slogans comme PASCAL PAS A PAS.

### PROBLÈME

Enoncé du problème

Préparation

21

24

### DISCUSSION

Premières étapes de l'analyse

Dessin des lettres

Développement

24

30

35

### RÉVISION

Survol des questions d'analyse et de développement

Retour sur la programmation

Synthèse de *La bannière est dans le SAC*

Lien avec l'étude de cas précédente

38

39

40

41

### SOURCES PASCAL

Un programme complet de *La bannière est dans le SAC*

42

## CHAPITRE 3 *La boutique de calendriers*

45

Garder la trace du temps est un problème majeur pour l'humanité. Au cours de l'histoire, les semaines et les mois ont été définis par des calendriers très différents de ceux que nous connaissons. Ce programme permet de créer un calendrier pour n'importe quelle année. Il peut être modifié pour créer des calendriers fantaisistes sans lundis ou recréer d'anciens calendriers avec des semaines de 10 jours ou des mois de 28 jours.

### PROBLÈME

Enoncé du problème

Préparation

Impression du premier semestre 1986

45

46

46

### DISCUSSION

Comprendre le programme

Analyse et programmation de `NbJoursDuMois`Analyse et programmation de `AfficherMois`

47

51

53

### RÉVISION

Survol des questions d'analyse et de développement

63

Retour sur la programmation	64
Synthèse de <i>La boutique de calendriers</i>	67
Lien avec les études de cas précédentes	69

## SOURCES PASCAL

Un cadre de programme inachevé	70
NbJoursDuMois	71
AfficherMois jour par jour	71
AfficherMois semaine par semaine	72

## CHAPITRE 4 *Construction d'une calculatrice romaine* 74

Les ordinateurs ont révolutionné les mathématiques; pensez que l'on apprenait le calcul des racines carrées à la main! Ce programme est celui d'une calculatrice qui donne l'équivalent décimal d'un nombre en chiffres romains.

## PROBLÈME

Enoncé du problème	74
Description des chiffres romains	74
Préparation	77

## DISCUSSION

Préliminaires	77
Méthode chiffre à chiffre	79
Solution globale	84
Autre utilisation d'un tableau	87

## RÉVISION

Survol des questions d'analyse et de développement	88
Retour sur la programmation	89
Synthèse de <i>Construction d'une calculatrice romaine</i>	90
Lien avec les études de cas précédentes	92

## SOURCES PASCAL

Méthode chiffre à chiffre	92
Solution globale	93
Variante utilisant un tableau des valeurs décimales des chiffres romains	94

## CHAPITRE 5 *Est-ce légal ?* 97

Comment les programmes se comportent-ils face à des données illégales? Supposer que les données seront entrées correctement est une source certaine de problèmes: par exemple quand l'utilisateur fait une faute de frappe - peut-être exprès, pour voir si le cas est prévu. Cette étude de cas traite de la protection contre les données illégales dans les programmes de *Vérifiez ce code !*, *La bannière est dans le SAC*, *La boutique de calendriers*, et *Construction d'une calculatrice romaine*.

PROBLÈME	
Enoncé du problème	97
Préparation	98
DISCUSSION	
Contrôler les entrées légales	98
Modification du programme de <i>Vérifiez ce code!</i>	102
Modification de <i>La bannière est dans le SAC</i> et de <i>La boutique de calendriers</i>	104
Une modification possible de <i>Construction d'une calculatrice romaine</i>	111
Une autre façon de modifier <i>Construction d'une calculatrice romaine</i>	122
RÉVISION	
Survol des questions d'analyse et de développement	131
Retour sur la programmation	133
Synthèse de <i>Est-ce légal?</i>	136
SOURCES PASCAL	
Variante du programme de <i>Vérifiez ce code!</i>	137
Version revue du programme de <i>La bannière est dans le SAC</i>	138
Variante pour la méthode de non-respect des règles	141
Variante pour la méthode de confirmation par les règles	144

## CHAPITRE 6 *Le texte prend l'espace*

149

Aujourd'hui les ordinateurs ont rendu possible la PAO - la publication assistée par ordinateur. Grâce aux logiciels de PAO, l'utilisateur peut présenter un texte presque comme dans un livre. Cette étude de cas décrit la construction d'une procédure qui pourrait être intégrée à un programme de traitement de texte pour justifier du texte entre ses marges gauche et droite.

PROBLÈME	
Enoncé du problème	149
Préparation	151
DISCUSSION	
Première approche	152
D'autres façons de concevoir <i>Le texte prend l'espace</i>	156
La méthode par-blanc	157
Le découpage par-intervalle	167
Amélioration du programme	171
RÉVISION	
Survol des questions d'analyse et de développement	174
Retour sur la programmation	175
Synthèse de <i>Le texte prend l'espace</i>	177

SOURCES PASCAL	
Le programme de test	179
La méthode par-blanc	181
Solution par-intervalle	182
Solution par-intervalle modifiée	183

## CHAPITRE 7 *Vous êtes ce que vous mangez* 185

L'enthousiasme de nos contemporains à adopter un régime de santé les conduit à noter les calories, les graisses et le cholestérol qu'ils consomment. Imaginons que l'on vous demande d'écrire un programme pour les aider. Cette étude de cas donne une idée de ce que pourrait faire un programmeur professionnel.

PROBLÈME	
Enoncé du problème	185
Préparation	187
DISCUSSION	
Découpage du programme	187
Résolution du problème	194
Mise au point du programme	205
RÉVISION	
Survol des questions d'analyse et de développement	207
Retour sur la programmation	208
Synthèse de <i>Vous êtes ce que vous mangez</i>	210
Lien avec les études de cas précédentes	212
SOURCES PASCAL	
Programme à mettre au point	212

## CHAPITRE 8 *Vous ne savez plus ce que vous avez mangé* 221

L'utilisateur du programme *Vous êtes ce que vous mangez* a plus de problèmes que prévu à se souvenir de ce qu'il mange. De plus il aimerait pour chaque aliment entrer une fois pour toutes les lipides et le nombre de calories. Cette étude de cas montre comment un programmeur professionnel prend en compte les demandes des utilisateurs. Et que veut exactement dire "convivial"?

PROBLÈME	
Enoncé du problème	221
Préparation	222
DISCUSSION	
Préparation des modifications	223
Mise en place des modifications dans <i>Vous êtes ce que vous mangez</i>	228

Analyse de haut-niveau de l'éditeur du journal	236
Implémentation des fonctionnalités	240
Fin de l'analyse et programmation de l'éditeur du journal	244
L'analyse de "obtenir la ligne, traiter le mot de commande"	249
L'analyse de "obtenir l'ensemble d'une commande légale"	259

#### RÉVISION

Survol des questions d'analyse et de développement	262
Retour sur la programmation	265
Synthèse de <i>Vous ne savez plus ce que vous avez mangé</i>	266
Lien avec les études de cas précédentes	268

#### SOURCES PASCAL

Révision de <i>Vous êtes ce que vous mangez</i>	268
Implémentation des opérations abstraites sur le fichier journal	273
Le reste de l'éditeur du journal	275
Retour sur l'approche de "Obtenir l'ensemble d'une commande légale"	284

## CHAPITRE 9 *Les défis des échecs*

289

Les programmes qui jouent aux échecs, résolvent ou créent des problèmes d'échecs sont très répandus. Il existe même un tournoi annuel mettant face à face des programmes, ou des maîtres et des programmes. Cette étude de cas traite un problème simple en rapport avec les échecs.

#### PROBLÈME

Thème	289
Énoncé du problème	290
Préparation	291

#### DISCUSSION

Comprendre le problème et choisir une implémentation	291
La solution reine-par-reine	292
La solution ligne-par-ligne	301
La représentation d'une position de reine	305

#### RÉVISION

Survol des questions d'analyse et de développement	310
Retour sur la programmation	311
Synthèse de <i>Les défis des échecs</i>	313
Lien avec les études de cas précédentes	314

#### SOURCES PASCAL

La solution reine-par-reine	315
La solution ligne-par-ligne	316
La représentation des positions des reines	320

## CHAPITRE 10 *Stratégie pour battre les cartes* 323

Tout le monde sait battre un jeu de cartes; le faire faire à un ordinateur demande quelques efforts. Est-ce que l'ordinateur doit imiter l'homme ou se comporter en machine ? Comment évaluer la qualité du résultat? Cette étude de cas montre des programmes appliquant différents algorithmes pour battre un jeu de cartes, et compare les résultats.

PROBLÈME	
Enoncé du problème	323
Préparation	324
DISCUSSION	
Comprendre le problème	324
Le "parfait batteur"	326
Battre avec un générateur de nombres aléatoires	330
Battre en faisant des échanges au hasard	331
Battre par tirages successifs	335
RÉVISION	
Survol des questions d'analyse et de développement	338
Retour sur la programmation	339
Synthèse de <i>Stratégies pour battre les cartes</i>	339
Lien avec les études de cas précédentes	341
SOURCES PASCAL	
BatteurParfait avec programme de test	342
BattreParSelection et sous-programmes	343
BattreParSelectionEfficace et version modifiée de PositionAleaDansJeu	344

## CHAPITRE 11 *Réparation d'une calculatrice romaine* 345

Comment savoir si un programme marche? Le Consortium Jules César propose un logiciel de conversion des chiffres romains en valeurs décimales. Vérifiez que leur programme marche, corrigez éventuellement les erreurs, et présentez votre note d'honoraires d'ingénieur-conseil.

PROBLÈME	
Enoncé du problème	345
Préparation	346
DISCUSSION	
Comprendre le problème	346
Test du programme	352
Une correction possible	355
RÉVISION	
Survol des questions d'analyse et de développement	359
Retour sur la programmation	360

Synthèse de <i>Réparation d'une calculatrice romaine</i>	362
Lien avec les études de cas précédentes	363

SOURCES PASCAL	
Solution du problème posé dans <i>Réparation d'une calculatrice romaine</i>	363
Un essai pour corriger la version boguée	364

## CHAPITRE 12 *Le club des mangeurs* 367

Quantité de gens veulent suivre leur alimentation et vérifier leurs progrès. Cette étude de cas étend les programmes de *Vous êtes ce que vous mangez* et de *Vous ne savez plus ce que vous avez mangé* pour traiter une liste compliquée d'aliments, des utilisateurs multiples, et plusieurs états de sortie. Le programme doit pouvoir trier des aliments ou d'autres informations, et le faire efficacement.

PROBLÈME	
Énoncé du problème	367
Préparation	369

DISCUSSION	
Améliorations de <i>Vous ne savez plus ce que vous avez mangé</i>	369
Amélioration de l'éditeur du journal et introduction d'un éditeur des aliments	380

RÉVISION	
Survol des questions d'analyse et de développement	383
Retour sur la programmation	383
Synthèse de <i>Le club des mangeurs</i>	384
Lien avec les études de cas précédentes	386

SOURCES PASCAL	
Dernières révisions du programme de <i>Vous êtes ce que vous mangez</i>	387

<i>Glossaire</i>	391
------------------	-----

<i>Paradigmes</i>	401
-------------------	-----

<i>Index</i>	451
--------------	-----