

X. Franch - J. Marco - X. Molinero
J. Petit - F. Xhafa

Fonaments de programació

Problemes resolts en C++

X. Franch - J. Marco - X. Molinero
J. Petit - F. Xhafa

Fonaments de programació

Problemes resolts en C++

Pròleg

El llibre *Fonaments de Programació* neix, com tantes altres vegades, quasi sense voler, a partir del material que hem anat elaborant els autors durant els anys que hem impartit docència a les assignatures d'Informàtica Bàsica i de Fonaments d'Informàtica als estudis de l'Escola Tècnica Superior d'Enginyeries Industrial i Aeronàutica de Terrassa (ETSE-IAT) i l'Escola Universitària d'Enginyeria Tècnica Industrial de Terrassa (EUETIT) de la Universitat Politècnica de Catalunya (UPC).

El nostre objectiu ha estat confeccionar una obra que s'adaptés a les característiques de les assignatures esmentades. Per això, hem intentat, en la mesura dels possibles, cobrir els aspectes bàsics i fonamentals en un curs d'introducció a la programació en el vessant aplicat. D'aquesta forma, volem solucionar parcialment l'absència de material específic de problemes per aquestes assignatures.

El resultat ha estat un text de exercicis d'autoavaluació, problemes i projectes resolts. Els exercicis solen ser enunciats curts i concisos, cadascun dels quals involucra una tècnica o un coneixement específic del temari. En general, es tracta d'exercicis que il·lustren els conceptes que s'han introduït a les classes de teoria. Els problemes resolts presenten una major complexitat que la dels exercicis i, freqüentment, involucren diferents conceptes teòrics que cal combinar per obtenir una bona solució. La majoria dels problemes han aparegut en exàmens parcials o finals recents en les assignatures de les titulacions esmentades (tot i que han estat adaptats i unificats). La resolució dels problemes permet mitigar les dificultats que sovint tenen els alumnes en aplicar sobre casos concrets els conceptes que han adquirit i, fins i tot, entès en les classes de teoria. Per això, hem intentat comentar exhaustivament les solucions per facilitar l'aprenentatge dels diversos mètodes de resolució dels problemes. Els projectes resolts pretenen ser un model per a la realització de projectes semblants, com els que els alumnes han de desenvolupar a les classes de laboratori.

El llibre utilitza un subconjunt bàsic del llenguatge de programació C++ per descriure els algorismes. Aquest subconjunt s'ha mantingut expressament tan petit com ha estat possible: només s'utilitza un paradigma de programació imperativa procedural i no es tenen en compte les capacitats d'orientació a objectes que ofereix aquest llenguatge (tot i que sí s'usen objectes per lectura, escriptura, cadenes de caràcters i taules).

Per fer funcionar aquests algorismes escrits en C++, el lector només haurà de reordenar els diferents fragments de cada programa i afegir-hi les capçaleres que, per raons de brevetat, no hem repetit a cada solució. En particular, només s'utilitzen unes poques funcionalitats de les llibreries `<iostream>`, `<string>`, `<vector>` i `<cmath>`.

Voldríem agrair als nostres companys del departament de Llenguatges i Sistemes Informàtics que han fet docència en aquestes assignatures la seva participació en la confecció d'alguns dels enunciats dels problemes que apareixen en aquest llibre, així com els seus comentaris sobre versions preliminars del material.

Esperem que el llibre sigui d'utilitat no només per als estudiants d'aquestes assignatures, sinó també per als d'altres assignatures similars que existeixen en centres no informàtics de la UPC i altres universitats de parla catalana.

Barcelona, juliol de 2006

Els autors

Índex

Pròleg	i
Índex	v
I Exercicis i problemes	1
1 Introducció	3
Exercicis	3
1.1 La congruència de Zeller per a calendaris	8
1.2 El valor clau per a calendaris	9
2 Seqüències	11
Exercicis	11
2.1 Misteri	14
2.2 El control de qualitat	14
2.3 Vaques boges	15
2.4 Infraccions	15
2.5 Codis de barres	15
2.6 El supermercat Fruita Madura	16
2.7 Productes caducats	16
2.8 El valor actual	16
2.9 La benzinera de Vilanova	17
2.10 El pàrquing de la Mútua de Terrassa	17
2.11 El polígon equilateral	18
2.12 La Pica d'Estats	19
2.13 Valls minimalis	19
2.14 El torneig de futbol	19

3 Accions i funcions	21
Exercicis	21
3.1 No ho sé	23
3.2 Tampoc no ho sé	23
3.3 Qui ho sap?	24
3.4 Incògnita	24
3.5 Comparar dates	25
3.6 Articles caducats	25
3.7 Intervals	25
3.8 Llaunes	25
3.9 Sumar dígits parells de $f(x)$	26
3.10 La pesta porcina	26
3.11 Leibniz i π	26
4 Taules	29
Exercicis	29
4.1 Sobre la taula	31
4.2 Pantans	32
4.3 L'Avi Pep	32
4.4 El pàrquing	33
4.5 Vols	33
4.6 Punt de creu	34
4.7 Vector suma columnes	34
4.8 Files i columnes perpendiculars	34
4.9 El valor propi	34
4.10 La planificació de tasques	35
4.11 La suma per capes	35
4.12 El monitor Monocrom TM	36
4.13 La codificació de missatges	37
4.14 Generació de permutacions	37
4.15 El segment nul més llarg	37
5 Tuples i estructures de dades	39
Exercicis	39
5.1 La farmacèutica de Sant Cugat	42
5.2 El parc mòbil d'Igualada	42
5.3 La biblioteca de Castelldefels	42
5.4 La Universitat de Mataró	43
5.5 L'associació de titulats	43
5.6 La xarxa de concessionaris	43
5.7 El museu de pintura	44
5.8 Províncies	44
5.9 L'hospital de Manresa	45
5.10 Polinomis	46
6 Disseny descendent	47
Exercicis	47
6.1 Departaments	48
6.2 La immobiliària de Canet de Mar	49
6.3 El control d'accés	50

6.4	Coixos F.C.	51
6.5	Vendes d'una empresa	52
6.6	El centre de traduccions Que diu que què	53
6.7	El quadrat màgic	55
6.8	El traductor automàtic	56
6.9	Els gratacels de Terrassa	57
7	Projectes	59
7.1	Borsa	59
7.2	Bàsquet	60
II	Problemes resolts	63
1	Introducció	65
1.1	La congruència de Zeller per a calendaris	65
1.2	El valor clau per a calendaris	66
2	Seqüències	67
2.1	Misteri	67
2.2	El control de qualitat	67
2.3	Vaques boges	68
2.4	Infraccions	68
2.5	Codis de barres	69
2.6	El supermercat Fruita Madura	70
2.7	Productes caducats	71
2.8	El valor actual	72
2.9	La benzinera de Vilanova	73
2.10	El pàrquing de la Mútua de Terrassa	74
2.11	El polígon equilatral	76
2.12	La Pica d'Estats	76
2.13	Valls minimalis	77
2.14	El torneig de futbol	78
3	Accions i funcions	81
3.1	No ho sé	81
3.2	Tampoc ho sé	81
3.3	Qui ho sap?	81
3.4	Incògnita	82
3.5	Comparar dates	82
3.6	Articles caducats	82
3.7	Intervals	83
3.8	Llaunes	84
3.9	Sumar dígit parells de $f(x)$	84
3.10	La pesta porcina	85
3.11	Leibniz i π	86

4	Taules	87
4.1	Sobre la taula	87
4.2	Pantans	88
4.3	L'Avi Pep	88
4.4	El pàrquing	89
4.5	Vols	90
4.6	Punt de creu	91
4.7	Vector suma columnes	91
4.8	Files i columnes perpendiculars	92
4.9	El valor propi	93
4.10	La planificació de tasques	94
4.11	La suma per capes	94
4.12	El monitor monocrom TM	96
4.13	La codificació de missatges	97
4.14	Generació de permutacions	99
4.15	El segment nul més llarg	101
5	Tuples i estructures de dades	105
5.1	La farmacèutica de Sant Cugat	105
5.2	El parc mòbil d'Igualada	107
5.3	La biblioteca de Castelldefels	108
5.4	La Universitat de Mataró	110
5.5	L'associació de titulats	111
5.6	La xarxa de concessionaris	112
5.7	El museu de pintura	113
5.8	Províncies	114
5.9	L'hospital de Manresa	115
5.10	Polinomis	116
6	Disseny descendent	119
6.1	Departaments	119
6.2	La immobiliària de Canet de Mar	120
6.3	El control d'accés	122
6.4	Coixos F.C.	125
6.5	Vendes d'una empresa	127
6.6	El centre de traduccions Que diu que què	128
6.7	El quadrat màgic	130
6.8	El traductor automàtic	132
6.9	Els gratacels de Terrassa	135
7	Projectes	137
7.1	Borsa	137
7.2	Bàsquet	147

Part I

Exercicis i problemes

1

Introducció

Exercicis

1. Penseu en totes les coses que vàreu fer ahir. En quines d'elles sou conscients d'haver usat un programa informàtic?
2. Expliqueu a un turista perdut com anar de la Plaça Major a l'Aeroport del Prat utilitzant transports públics en una dia de vaga de taxis.
3. Expliqueu com multiplicar dos nombres a un nen que sap sumar i coneix les taules de multiplicar.
4. Què és un algorisme?
5. Digueu quin és el contrari de les afirmacions següents:
 - a és més petit que b .
 - Plou i fa sol.
 - No plou i no fa sol.
 - Dues rectes tenen un punt en comú.
 - Tots els camins porten a Roma.
 - Tots els ànecs són blancs i volen.
 - P i Q .
 - P o Q .
 - P o Q però no les dues.
6. Feu les taules de veritat de **no** (a i b) i de **no** a o **no** b . Quina és la vostra conclusió? Feu el mateix per **no** (a o b) i per **no** a i **no** b .
7. Feu les taules de veritat de **no** (a i b i c) i de **no** a o **no** b o **no** c . Quina és la vostra conclusió? Feu el mateix per **no** (a o b o c) i per **no** a i **no** b i **no** c .
8. Epimènides de Creta va afirmar: "Tots els cretencs són uns mentiders". Què en penseu?

9. Representeu els enters 0, -1, 73, 81 i -524 en una màquina que té les paraules de 16 bits usant l'estratègia de representació per signe i magnitud. Quin és l'interval d'enters que es pot representar en aquesta màquina?
10. En el codi ASCII, la 'a' es representa amb el valor numèric 97, la 'A' amb el 65 i el '0', amb el 48. Digueu els valors numèrics que representen els caràcters '3', '8', 'M' i 'Z' en aquest codi. Representeu aquests valors numèrics usant paraules de 8 bits.
11. Quins són els tipus bàsics de C++? Digueu quins valors són admissibles per a cada tipus. Feu una taula on per a cada operador llisteu quins són els tipus dels seus operands i quin és el tipus del seu resultat. Assenyaleu quins errors es poden donar en aplicar aquests operadors.
12. Digueu de quin tipus són les expressions següents i avalueu-les:

- 12
- 12.0
- 3 + 4 * 5
- 8 / 4
- 8 / 3
- 8.0 / 3.0
- 2 == 3
- 3 >= 2
- (3 % 2 == 0) == (7 / 2 > 2)
- 3 + (4 * 5)
- 4 / (8 / 10)
- *true and false or true*
- (3 + 4) * 5
- (((3 + 4) * 5))
- (4 / 8) / 10
- *true and (false or true)*
- *not ('S' < 'D')*
- (5 + 1) / (9 % 3)
- *double(34)*
- *int(4.9) + int(4.1)*
- *int(4.9 + 4.1)*
- *int('9')*
- *char(65)*
- *int('2') >= int('9')*
- *char(65)*
- *a / 0*

13. Donades aquestes declaracions:

```
int i, j, k;
double x, y, z;
char a, b, c;
bool p, q, r;
```

digueu quines de les expressions següents són inadmissibles en C++ i per què:

- a
- i+3 <= x
- (j-k) / i
- (j-k) / x

- $(j-k) / \text{int}(x)$
- $(i * j - z >= x * y * y) == (\text{not } a \text{ and } b)$
- $|x - y| + z$
- $\sqrt{x} > 0$
- $x^2 - x^2$
- $\text{not } (p \text{ and } q \text{ and } r \text{ and } c >= 'A')$
- $(p+r)+i$
- $3*i-(j*z)*x$
- $a >= b \text{ and } b > c$
- $(3*x) / (3.1416*y)$
- $\frac{x+y+z}{i}$

14. Expliqueu el significat de les instruccions d'assignació, de seqüenciació, alternativa, iterativa, de lectura i d'escriptura.
15. Expliqueu com es comporten els operadors de divisió entera i mòdul ($/$ i $\%$) amb operands negatius. Calculeu $7 / 4$, $-7 / 3$, $7 / -3$, $-7 / -3$, $7 \% 3$, $-7 \% 3$, $7 \% -3$, i $-7 \% -3$.
16. Donades aquestes inicialitzacions:

```
int z = 0;
int u = 1;
double x = 3.5;
bool t = true;
```

avalueu les expressions següents i digueu quines poden produir problemes:

- $\text{not } (z <= 1 \text{ and } x >= 0 \text{ and } t)$
- $z == 0 \text{ and } x <= 0 \text{ or } u == 1 \text{ and } x >= 0$
- $u / z > -2$
- $z != 0 \text{ and } u / z > -2$
- $u / z > -2 \text{ and } z != 0$
- $z == 0 \text{ or } u / z > -2$
- $u / z > -2 \text{ or } z == 0$
- $z == 0 \text{ or } 24 \% z != 1$
- $z == 0 \text{ and } 24 \% z != 1$
- $x == 3.5$

17. Dissenyeu un algorisme que llegeixi dos nombres reals i escrigui la seva suma.
18. Dissenyeu un algorisme que llegeixi un nombre enter i escrigui si és parell o senar.
19. Dissenyeu un algorisme que llegeixi un nombre enter entre 100 i 999 i escrigui si és cap-i-cua.
20. Dissenyeu un algorisme que, donats tres enters que representen hores, minuts i segons, doni l'equivalent en segons.
21. Dissenyeu un algorisme que, donats tres enters que representen hores, minuts i segons, sumi un segon i doni el resultat en el mateix format.
22. Dissenyeu un algorisme que, donada una quantitat de segons, digui quantes hores, minuts i segons representa.

23. Dissenyeu un algorisme que donada una quantitat en euros, doni el nombre mínim de bitllets i monedes necessaris, sabent que es disposa de monedes de 1, 2, 5, 10, 20 i 50 cèntims i 1 i 2 euros, i bitllets de 5, 10, 20, 50, 100, 200 i 500 euros.
24. Escriviu expressions que, donat un real r , calculin
- la part entera per defecte de r ,
 - la part entera per excés de r ,
 - el valor enter més proper a r .
25. Escriviu expressions booleanes que, donat un caràcter c , indiquin
- si c és una lletra minúscula,
 - si c és una lletra majúscula,
 - si c és un dígit,
 - si c no és ni minúscula ni majúscula ni dígit,
 - si c és una vocal,
 - si c és una consonant,
 - si c és un símbol de puntuació.
26. Escriviu una expressió entera que es correspongui al valor d'un dígit donat (caràcters de '0' a '9').
27. Escriviu una expressió que converteixi un enter entre 0 i 9 al seu dígit corresponent.
28. Dissenyeu un algorisme que donats dos intervals reals $[a_1, b_1]$ i $[a_2, b_2]$ digui si el primer es troba dins del segon.
29. Dissenyeu un algorisme que donats dos intervals reals $[a_1, b_1]$ i $[a_2, b_2]$ calculi l'interval corresponent a la seva intersecció o indiqui que és buida.
30. Dissenyeu un algorisme que donades dues hores del rellotge $h_1:m_1:s_1$ i $h_2:m_2:s_2$, indiqui si una altra hora $h:m:s$ es troba entre la primera i la segona.
31. Per als problemes següents, identifiqueu quines són les entrades, quines són les sortides, quina relació hi ha entre les entrades i les sortides, quines condicions han de complir les entrades i quins errors es poden donar. No feu els algorismes:
- Calcular la suma de dos nombres reals.
 - Calcular el producte de dos nombres reals.
 - Calcular el quocient de dos nombres reals.
 - Calcular el quocient i el residu de dos nombres enters.
 - Calcular el valor absolut d'un nombre real.
 - Calcular l'arrel quadrada d'un nombre real.
 - Resoldre una equació lineal.
 - Resoldre una equació de segon grau.
 - Trobar el valor més alt d'una llista d'enters.
 - Trobar la mitjana de les notes dels alumnes d'una classe.
 - Calcular la distància entre dos punts del pla.
 - Calcular la distància entre dos punts de l'espai.
 - Esbrinar si dues línies són iguals, paral·leles i diferents, o s'intersecten.
 - Simplificar una fracció.
 - Decidir si dues fraccions representen el mateix nombre racional.

32. Un any de traspàs és un any civil que consta de 366 dies civils. Són anys de traspàs tots els múltiples de 4 que no ho són de 100, o que són de 100 però també de 400. Escriuiu una expressió que indiqui si una variable entera *any* és de traspàs o no.
33. Dissenyeu un algorisme que llegeixi dos enters i n'escrigui el màxim.
34. Dissenyeu un algorisme que llegeixi dos enters i els escrigui ordenats.
35. Dissenyeu un algorisme que llegeixi tres enters i els escrigui ordenats.
36. Dissenyeu un algorisme que llegeixi quatre enters i els escrigui ordenats.
37. Dissenyeu un algorisme que llegeixi un enter que representa una data en format DDMMAAAA i escrigui el dia, el mes i l'any corresponent.
38. Dissenyeu un algorisme que llegeixi un enter que representa una hora del rellotge en format HHMMSS i escrigui l'hora, els minuts i els segons corresponents.
39. Dissenyeu un algorisme que llegeixi dues dates (dia, mes i any de cadascuna) i digui si la primera és anterior en el temps a la segona.
40. Dissenyeu un algorisme que digui si un enter positiu és primer o no.
41. Dissenyeu un algorisme que escrigui el producte de factors primers d'un número natural.
42. Dissenyeu un algorisme que escrigui els 10 primers números primers de la forma $2^n - 1$ amb n natural.
43. Dissenyeu un algorisme que escrigui les taules de multiplicar del 0 al 9.
44. El teorema de Fermat estableix que l'equació $a^n + b^n = c^n$ no té solució per a a , b i c enters diferents de 0, amb $n \geq 3$. Feu un algorisme que comprovi la certesa del teorema per a $1 \leq a, b \leq 1000$ i $3 \leq n \leq 100$.
45. Un dels problemes potencials de les operacions aritmètiques en els ordinadors és el sobreiximent: el resultat d'una operació no es pot representar amb la paraula de l'ordinador perquè falten bits. Un algorisme totalment fiable hauria sempre de verificar que una operació aritmètica no provoca sobreiximent. Dissenyeu un algorisme que llegeixi dos enters i , per a cadascuna de les operacions aritmètiques de suma, resta, producte, divisió i mòdul detecti si provoca sobreiximent o no aplicada sobre aquests dos enters.
46. Dissenyeu un algorisme que intercanviï el valor de dues variables enteres sense usar cap variable auxiliar.
47. Què penseu de l'algorisme següent?

```
int x = ...
int i = 10
while (i > 0) {
    if (x == x) i = i + 1; else i = i - 1;
}
```

48. Què penseu de l'algorisme següent?

```
int i = ...;
while (i != 0) {
    i = i - 2;
}
```

49. Simplifiqueu l'algorisme següent:

```
int x = ..., y = ...;
bool iguals;
if (x == y) iguals = true; else iguals = false;
```

50. Un mag diu: “Pensa’t un número. Multiplica’l per dos. Suma-li 34. Divideix-lo per dos. Treu-li el número que havies pensat. Te’n queden 17!”

Què en penseu?

1.1 La congruència de Zeller per a calendaris

La congruència de Zeller és un càlcul que permet obtenir el dia de la setmana per a una data qualsevol del calendari gregorià. Donada una data determinada pel triplet $\langle d, m, a \rangle$, on d és el dia del mes, m és el mes de l’any i a és l’any,

1. Se li resten dues unitats al mes m i si dóna zero o menys se li suma 12 al mes i se li resta una unitat a l’any. El nou mes obtingut l’anomenem m' i el nou any a' .
2. Es calcula la centúria c (els dos primers dígit de l’any) a partir de l’any a' .
3. Es calcula l’any dins la centúria y (els dos darrers dígit de l’any) a partir de l’any a' .
4. Es calcula

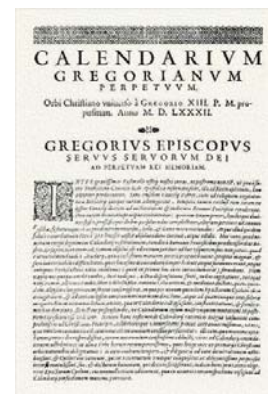
$$f = [2.6m' - 0.2] + d + y + [y/4] + [c/4] - 2c,$$
 on $[x]$ és la part entera per defecte de x .
5. f mòdul 7 representa el dia de la setmana segons la correspondència següent: 0 = diumenge, 1 = dilluns, ...

Utilitzeu la congruència de Zeller per determinar en quin dia de la setmana va néixer. Funciona?

Dissenyeu un algorisme que, tot utilitzant la congruència de Zeller, llegeixi una data (dia, mes i any) i escrigui en quin dia de la setmana cau.

1.2 El valor clau per a calendaris

Un any de traspàs és un any civil que consta de 366 dies civils. Després de la reforma gregoriana són anys de traspàs tots els anys múltiples de quatre que no acabin en dos zeros i també tots aquells que, acabats en dos zeros, tinguin el nombre que quedaria en treure els dos zeros final divisible per quatre. Així, 1700, 1800, 1900, tot i ésser múltiples de 4, no foren de traspàs; en canvi, l'any 2000 ho va ser.



El mètode del *valor clau* és un càlcul que permet obtenir el dia de la setmana per a una data qualsevol en el calendari gregorià. Donat un dia determinat pel triplet $\langle d, m, a \rangle$, on d és el dia del mes, m és el mes de l'any i a és l'any,

1. Es calcula la centúria c (els dos primers dígits de l'any) a partir de l'any a .
2. Es calcula l'any dins la centúria y (els dos darrers dígits de l'any) a partir de l'any a .
3. S'agafa el residu r de dividir c entre 4.

4. Es calcula

$$v_1 = \left\lfloor \frac{y}{4} \right\rfloor + d + 11 - 2r + y$$

on $\lfloor x \rfloor$ és la part entera per defecte de x .

5. Es calcula

$$v_2 = v_1 + \text{clau}(m)$$

on $\text{clau}(m)$ és una funció que pren els valors següents:

m	1	2	3	4	5	6	7	8	9	10	11	12
$\text{clau}(m)$	1	4	4	0	2	5	0	3	6	1	4	6

6. Si el mes m correspon a gener o febrer i l'any a és de traspàs, es resta 1 a v_2 .
7. Es calcula s , que val el resultat d'agafar el residu de dividir v_2 entre 7 i afegir-li 1.
8. El valor de s indica el dia de la setmana segons la correspondència següent: 1 = dilluns, 2 = dimarts, ...

Utilitzeu el mètode del valor clau per determinar en quin dia de la setmana va néixer. Funciona?

Dissenyu un algorisme que, tot utilitzant el mètode del valor clau, llegeixi una data (dia, mes i any) i escrigui en quin dia de la setmana cau.

Seqüències

Exercicis

1. Dissenyeu un programa que llegeixi una seqüència d'enters acabada per un zero i digui quants enters la componen.
2. Dissenyeu un programa que llegeixi una seqüència d'enters acabada per un zero i compti quants cops apareix el número 123.
3. Dissenyeu un programa que llegeixi una seqüència d'enters acabada per un zero i compti quants números positius hi apareixen.
4. Dissenyeu un programa que llegeixi una seqüència d'enters acabada per un zero i digui si hi ha més positius que negatius.
5. Dissenyeu un programa que llegeixi una frase acabada per un punt i digui quants caràcters la componen.
6. Dissenyeu un programa que llegeixi una frase acabada per un punt i digui quants cops hi apareix la lletra Q.
7. Dissenyeu un programa que llegeixi una frase acabada per un punt i digui quantes vocals hi apareixen.
8. Dissenyeu un programa que llegeixi una frase acabada per un punt i digui si hi ha més vocals que consonants.
9. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i en calculi la mitjana aritmètica.
10. Donada una frase acabada per un punt, dissenyeu un algorisme que compti quantes paraules té.
11. Donada una frase acabada per un punt, dissenyeu un algorisme que compti quantes paraules tenen la mateixa longitud que la primera.

12. Donada una frase acabada per un punt, dissenyeu un algorisme que compti quantes vegades apareix la primera paraula.
13. Donada una frase acabada per un punt que no pot tenir blancs a l'inici, compteu el nombre de lletres que tenen la primera i la darrera paraula. Considereu que les paraules se separen per blancs i que no hi ha símbols de puntuació.
14. Donada una frase acabada per un punt que pot tenir blancs a l'inici, compteu el nombre de lletres que tenen la primera i la darrera paraula.
15. Donada una frase acabada per un punt, digueu si totes les paraules comencem amb la mateixa lletra.
16. Donada una frase acabada per un punt, digueu si totes les paraules acaben amb la mateixa lletra.
17. Donada una frase acabada per un punt, compteu totes les paraules que només contenen una sola vocal.
18. Dissenyeu un programa que, donat un enter positiu, digui quantes xifres té.
19. Dissenyeu un programa que generi els 100 primers termes de la sèrie de Fibonacci, definida com

$$F(n) = \begin{cases} 0, & n = 0, \\ 1, & n = 1, \\ F(n-1) + F(n-2), & n \geq 2. \end{cases}$$
20. Dissenyeu un programa que escrigui del revés la representació binària d'un enter positiu, usant el menor nombre de bits possible. Per exemple, per a l'enter 18, hauria d'escriure 01001.
21. Dissenyeu un programa que escrigui del revés la representació binària d'un enter positiu, usant el mínim nombre de bits múltiple de 8 possible. Per exemple, per a l'enter 18, hauria d'escriure 01001000.
22. Dissenyeu un programa que llegeixi una sèrie de zeros i uns, corresponent a la representació binària d'un enter positiu vista del revés, i digui quin és l'enter representat.
23. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui quins són l'element més petit i el més gran.
24. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui la diferència entre l'element més petit i el més gran.
25. Dissenyeu un programa que llegeixi una frase acabada per un punt i digui el nombre de vegades que apareix la síl·laba LA.
26. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si hi ha algun element més gran que 512.
27. Dissenyeu un programa que llegeixi una frase acabada en un punt i digui si hi ha alguna lletra R.
28. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si un element donat hi apareix.

29. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i ordenada creixentment i digui si un element donat hi apareix.
30. Dissenyeu un programa que llegeixi una frase acabada per un punt i digui si totes les vocals hi apareixen alguna vegada.
31. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si la diferència entre l'element més petit i el més gran és més gran o igual que 69.
32. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si és creixent o no.
33. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si conté dos elements consecutius repetits.
34. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si conté tres elements consecutius repetits.
35. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si formen una progressió aritmètica o no.
36. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i compti el nombre de pics estrictes. Un pic estricte apareix quan un element és estrictament més gran que el seu anterior i el seu successor en la seqüència. El primer i l'últim elements no poden ser pics estrictes. Per exemple, és un pic estricte el tros de seqüència 4 6 5, mentre que no ho és el tros de seqüència 4 6 6 5.
37. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si té algun pic estricte.
38. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si té algun pic (no necessàriament estricte). Un pic (no necessàriament estricte) apareix quan hi ha diversos elements iguals que són estrictament més grans que el seu anterior i el seu successor en la seqüència. El primer i l'últim elements no poden ser pics. Per exemple, és un pic el tros de seqüència 4 6 6 6 6 5.
39. Dissenyeu un programa que llegeixi una seqüència d'enters positius acabada per un zero i digui si hi ha més pics estrictes que valls estrictes. La definició de vall estricta és simètrica a la de pic estricte. Així, el tros de seqüència 6 3 7 forma una vall estricta, però 6 3 3 7 no.
40. Determineu el primer nombre natural n que compleix la propietat $17n + 9 = 2n^2$.
41. Malgrat les protestes de les organitzacions ecologistes, la companyia PetiQuiPeti ha instal·lat una central nuclear i ha decidit incorporar-hi un control de qualitat basat en el mostreig periòdic dels nivells de radioactivitat. Dissenyeu un programa que llegeixi una seqüència d'enters que representa el nivell diari de radioactivitat i que escrigui un '*' cada vegada que detecti una entrada amb valor superior o igual a 25, o bé dues entrades seguides superiors o iguals a 15. El programa acabarà o bé en detectar un nivell -1, que indica que la central nuclear ha tancat, o en detectar un nivell 50, que representa un nivell de radioactivitat tan gran que assegura que no queda ningú a qui avisar.

42. Dissenyau un programa que indiqui si una expressió acabada en un ‘#’ està ben parentitzada. Es considera que una expressió està ben parentitzada si es compleix que:

- En qualsevol moment de la lectura de l’expressió es té que el nombre de parèntesis oberts és més gran o igual al nombre de parèntesis de tancar.
- Un cop s’ha acabat de llegir l’expressió, tenim el mateix nombre de parèntesis oberts que de tancats.

Per exemple, “))a+b)((#” està mal parentitzada, però “(a+b)((a+b))#” està bé.

43. Apliqueu certes regles bàsiques de correcció ortogràfica a un text escrit en castellà que acaba per un caràcter ‘*’. Concretament, heu de donar com a sortida el mateix text tot aplicant les regles següents:

- Les paraules formades per una sola ‘i’ han de canviar-se a ‘y’.
- Cal canviar les ‘n’ que hi ha davant de ‘p’ per ‘m’ (conpra → compra).
- Cal canviar les síl·labes ‘nv’ per ‘mb’ (canviar → cambiar).
- Les ‘q’ sempre han d’anar seguides per ‘u’ (qeso → queso).
- La primera lletra que hi hagi darrera d’un punt sempre ha d’estar en majúscula.

2.1 Misteri

Considerem el programa següent:

```
int main() {
    int a;
    cin >> a;
    int m = a;
    while (a > 0) {
        if (m < a) m = a;
        cin >> a;
    }
    cout << m << endl;
}
```

- a) Quina és la sortida d’aquest programa donada l’entrada 3 8 9 4 2 -1?
- b) Descriu en una frase què fa aquest programa.

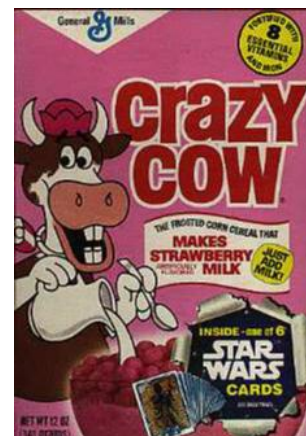
2.2 El control de qualitat

La cadena de producció d’una fàbrica de components electrònics vol dissenyar un programa per al control de la qualitat de les peces que fabriquen. Per poder sortir al mercat, les peces han de tenir un pes entre 100 i 150 grams. Per això, l’empresa vol que dissenyem un programa que donada la seqüència dels pesos (en grams) d’una partida de peces detecti si totes passen el control de qualitat. La seqüència acaba amb un pes fictici -1 que es fa servir com a sentinella.

2.3 Vaques boges

Temps de vaques boges... A fi d'evitar que entrin vaques boges a la cadena alimentària, la Generalitat de Catalunya ha decidit posar un codi (nombre enter) a cada vaca per poder distingir vaques malaltes (boges) i vaques sanes. Els codis s'escullen de tal manera que si una vaca és boja llavors el seu codi és múltiple de 13 i de 15. Quan arriba un grup de vaques a l'escorxador per ser sacrificades, un operari introdueix a l'ordinador els codis de totes les vaques i al final posa un 0 com a codi fictici.

Dissenyeu un programa que indiqui si hi ha vaques boges en un grup de vaques.



2.4 Infraccions

La zona de l'Eixample de Barcelona pateix des de fa temps problemes en les zones de càrrega i descàrrega per les infraccions que hi fan els vehicles de transport. Bàsicament, s'han rebut queixes que els vehicles superen el temps reglamentari de càrrega/descàrrega. Per aquest motiu, l'Ajuntament de Barcelona ha instal·lat uns sensors en aquestes zones que cada vegada que surt un vehicle envia a un ordinador central l'hora d'entrada i l'hora de sortida del vehicle (en minuts), suposant que estem dins el mateix dia.

Dissenyeu un programa que, donada la seqüència de dades que envien els sensors a l'ordinador central en un dia, acabada amb hores fictícies $\langle -1, -1 \rangle$ (per l'hora d'entrada i de sortida, respectivament), calculi:

- El nombre total d'infraccions que hi ha hagut al llarg del dia, és a dir, el nombre de vehicles que han superat el temps d'estada màxim, que actualment és de 30 minuts.
- El temps d'estada mitjana.

2.5 Codis de barres

Actualment la majoria de les empreses fan servir els codis de barres per distingir els seus productes. El codi més estès i que tendeix a convertir-se en un estàndard és l'EAN-13. Aquest codi està format per 13 dígits: 12 que es corresponen al codi del producte seguits d'un dígit de control. El dígit de control és una part molt important de la codificació EAN, perquè possibilita l'eliminació total dels errors de lectura del codi. Concretament, el dígit de control es calcula en dues fases:



- Se sumen els dígits que ocupen una posició senar multiplicats per 3 i els que ocupen una posició parella (els dígits es numeren de dreta a esquerra).
- Es resta el resultat de la suma a la desena superior o igual més propera a aquest resultat.

Per exemple, el dígit de control corresponent al codi 54490000099 es calcularia:

$$a) 9 * 3 + 9 + 0 * 3 + 0 + 0 * 3 + 0 + 0 * 3 + 0 + 9 * 3 + 4 + 4 * 3 + 5 = 84$$

Part II

Problemes resolts

1

Introducció

1.1 La congruència de Zeller per a calendaris

La resolució d'aquest problema involucra tres etapes:

1. Llegir les dades d'entrada (una data, composta d'un dia d , un mes m i un any a).
2. Realitzar els cinc passos descrits a l'enunciat, tot utilitzant variables auxiliars.
3. Escriure el resultat.

Aquesta és la implementació completa:

```
#include <iostream>
using namespace std;

int main () {
    // Llegir les dades
    int d, m, a;  cin >> d >> m >> a;

    // Calcular quin dia és de la setmana amb la congruència de Zeller
    int a2, m2;
    if (m-2 <= 0) {
        m2 = m - 2 + 12;  a2 = a - 1;
    } else {
        m2 = m - 2;      a2 = a;
    }
    int c = a2 / 100;
    int y = a2 % 100;
    int f = (26*m2-2)/10 + d + y + (y/4) + (c/4) - 2*c;
    int s = f%7;
    if (s < 0) s += 7;          // Compte!

    // Escriure el dia de la setmana corresponent
    if (s == 0) cout << "Diumenge" << endl;
```

```

else if (s == 1) cout << "Dilluns" << endl;
else if (s == 2) cout << "Dimarts" << endl;
else if (s == 3) cout << "Dimecres" << endl;
else if (s == 4) cout << "Dijous" << endl;
else if (s == 5) cout << "Divendres" << endl;
else          cout << "Dissabte" << endl;
}

```

Observeu que no cal controlar que `s` sigui 6 per escriure `Dissabte`. Fixeu-vos també que el mòdul de C++ no va bé amb els negatius i que cal tenir-ho en compte.

1.2 El valor clau per a calendaris

La solució d'aquest problema segueix la mateixa idea que l'anterior:

```

#include <iostream>
using namespace std;

int main () {
    // Llegir la data
    int d, m, a;  cin >> d >> m >> a;

    // Calcular el dia de la setmana amb el valor clau
    int c = a / 100;
    int y = a % 100;
    int r = c % 4;
    int v1 = y/4 + d + 11 - 2*r + y;
    int v2;
        if (m == 1 or m == 10)          v2 = v1 + 1;
    else if (m == 2 or m == 3 or m == 11) v2 = v1 + 4;
    else if (m == 5)                    v2 = v1 + 2;
    else if (m == 6)                    v2 = v1 + 5;
    else if (m == 8)                    v2 = v1 + 3;
    else if (m == 9 or m == 12)         v2 = v1 + 6;
    else                                 v2 = v1 + 0;

    bool traspas = a%400 == 0 or a%4 == 0 and a % 100 != 0;
    if (m <= 2 and traspas) v2 = v2 - 1;
    int s = v2%7 + 1;

    // Escriure el dia corresponent
        if (s == 7) cout << "Diumenge" << endl;
    else if (s == 1) cout << "Dilluns" << endl;
    else if (s == 2) cout << "Dimarts" << endl;
    else if (s == 3) cout << "Dimecres" << endl;
    else if (s == 4) cout << "Dijous" << endl;
    else if (s == 5) cout << "Divendres" << endl;
    else          cout << "Dissabte" << endl;
}

```

2

Seqüències

2.1 Misteri

- a) El programa escriu 9.
- b) Aquest programa calcula el màxim d'una seqüència d'enters acabada per un nombre negatiu (si la seqüència és buida, escriu el finalitzador).

2.2 El control de qualitat

Per resoldre aquest problema s'ha d'aplicar l'esquema de cerca a la seqüència de pesos de les peces

$$pes_1 pes_2 \dots pes_k \dots - 1$$

ja que es cerca una peça que no passi el control de qualitat. La forma d'obtenir els elements de la seqüència és:

- *⟨primer element⟩*: **llegir**(*pes*₁)
- *⟨següent element⟩*: **llegir**(*pes*_{*k*+1})
- *⟨darrer element⟩*: *pes*_{*k*} = -1

Ara dissenyem el programa tenint en compte que la propietat de cerca és que el pes actual no estigui entre 100 i 150 grams, que són respectivament el *pesMinim* i el *pesMaxim* que pot tenir una peça:

```
const int PES_MINIM = 100;
const int PES_MAXIM = 150;

int main() {
    bool trobat = false;
    int pes;    cin >> pes;                // Obtenir primer element
    while (pes != -1 and not trobat) {    // Mirar si és darrer element
```

```

    if (pes < PES_MINIM or pes > PES_MAXIM) { // Mirar condició de cerca
        trobat = true;
    } else {
        cin >> pes; // Obtenir següent element
    }
}
if (trobat) cout << "NO" << endl;
else cout << "SI" << endl;
}

```

Noteu que en cas de trobar una peça que no passa el control de qualitat la resposta ha de ser “NO” i que en cas de no trobar-ne cap la resposta ha de ser “SI”.

2.3 Vaques boges

Primer observem que darrera el problema a resoldre hi ha una seqüència d'enters acabada amb el sentinella 0 (codi fictici). Aquesta seqüència es defineix com a

$$\text{codi}_1 \text{ codi}_2 \dots \text{codi}_k \dots 0$$

i la manera d'obtenir els seus elements és:

- *⟨primer element⟩*: Llegir primer codi (**llegir**(codi₁)).
- *⟨següent element⟩*: Llegir següent codi (**llegir**(codi_{k+1})).
- *⟨darrer element⟩*: Mirar si el codi actual és 0 (codi_k = 0).

Observem ara que per a la solució del problema cal aplicar l'esquema de cerca sobre la seqüència, tot buscant un element de la seqüència que és múltiple de 13 i de 15. Aquesta propietat la podem expressar en C++ amb l'expressió: `codi % 13 == 0 and codi % 15 == 0`:

```

int main() {
    bool trobat = false;
    int codi;    cin >> codi; // Obtenir primer element
    while (codi != 0 and not trobat) { // Mirar si és darrer element
        if (codi%13 == 0 and codi%15 == 0) { // Mirar condició de cerca
            trobat = true;
        } else {
            cin >> codi; // Obtenir següent element
        }
    }
    if (trobat) cout << "Alerta: Hi ha vaques boges." << endl;
    else cout << "Calma: No hi ha vaques boges." << endl;
}

```

2.4 Infraccions

El programa que hem de dissenyar s'obté aplicant l'esquema de recorregut sobre la seqüència dels elements *⟨he, hs⟩* que es llegeix tot observant que l'element sentinella és *⟨−1, −1⟩*. La manera d'obtenir els elements de la seqüència és:

- *<primer element>*: `llegir(he1, hs1)`
- *<següent element>*: `llegir(hek+1, hsk+1)`
- *<darrer element>*: `hek = -1 i hsk = -1`

Quant al tractament de l'element de la seqüència, hem de fer:

- Comptar un vehicle més (a fi de calcular el temps d'estada mitjana).
- Sumar el temps d'estada del vehicle, o sigui la diferència entre l'hora de sortida i l'hora d'entrada, al temps total d'estada (a fi de calcular el temps d'estada mitjana). Noteu que podem restar les hores de sortida i entrada directament ja que vénen donades en minuts (nombres enters).
- Comprovar si el temps d'estada del vehicle supera els 30 minuts permesos i, en cas afirmatiu, comptar una infracció més.

Apuntem finalment que el temps d'estada mitjana es calcularà en fer el tractament final de la seqüència dividint el temps total d'estada pel nombre de vehicles. Cal doncs evitar divisions per zero (cosa que pot ocórrer si el nombre de vehicles és zero, és a dir, si la seqüència és buida):

```
const int TEMPS_MAXIM = 30;

int main() {
    // Tractament inicial
    int ni = 0;    // Nombre d'infraccions
    int nv = 0;    // Nombre de vehicles
    int tet = 0;   // Temps d'estada total
    // Llegir el primer element de la seqüència
    int he, hs;   cin >> he >> hs;
    // Recorregut
    while (he != -1 or hs != -1) {
        // Tractar l'element en curs
        ++nv;
        int te = hs - he; // Temps d'estada del vehicle actual
        tet = tet + te;
        // Comprovar si és una infracció
        if (te > TEMPS_MAXIM) ++ni;
        // Llegir l'element següent
        cin >> he >> hs;
    }
    if (nv != 0) {
        cout << "Nombre d'infraccions: " << ni << endl
             << "Temps d'estada mitjana: " << double(tet) / double(nv) << endl;
    } else cout << "No hi havia vehicles" << endl;
}
```

2.5 Codis de barres

Apliquem l'esquema de recorregut sobre les xifres que componen el nombre enter donat, per després calcular el dígit de control. Cal, doncs, definir la seqüència de les xifres d'un