þ7 #
$ ¼ ¼ ¼ ò Ð¾ Ð¾ Ð¾ ÐÎ v ÑÐ : Ñ~ Ñ~
š Ò x ½ ¾ Ò Ò° ÖÀ * Öê„ ¼
ÖÀ ÒÊ ö ÖÀ ÖÀ [I ÖÀ ÖÀ ÖÀ ÖÀ ÖÀ
ÖÀ

INGRID: A Software Tool for Extremal Graph Theory Research

A Primer and User's Manual

Ronald D. Dutton, Department of Computer Science

and

Robert C. Brigham, Department of Computer Science and
Department of Mathematics

University of Central Florida
Orlando, Florida 32816

ABSTRACT

INGRID, for INteractive GRaph Invariant Delimiter, is a computer program which allows a user to specify a value or range of values for one or more of up to 37 graph invariants. Embedded in the system is a data base of over 410 theorems which are selectively invoked and give the user values and/or ranges of values for the remaining graph invariants. The system is then poised to accept further invariant specifications, to cancel the effects of the last input, or to start over from the beginning. Facilities also exist which aid the user in determining the sequence of theorems causing some particular result. Although INGRID is not a 'theorem proving' program, it has pointed out relationships which are not part of its data base and at least some of these have not been found in the literature.

INTRODUCTION

An invariant of a graph is a property which remains constant under all labelings of the graph's nodes and edges and under all ways of drawing the graph. Such properties are as simple as the number of nodes and number of edges and as complex as the chromatic
number and maximum clique size. Extremal graph theory involves

the study of interrelationships between invariants. In the literature there are perhaps several dozen such invariants and several hundred interrelationship theorems. INGRID (INteractive GRaph Invariant Delimiter) is an attempt, for a specified subset of invariants, to collect and place in an integrated software package as many relevant theorems as possible. It allows a user to interactively assign a value or range of values to one or more of the invariants and to immediately see the effects upon other invariants as determined by the collected theorems. The utility of such an endeavor is, of course, limited by the choice of invariants and the scope of the theorems included in the system. The set of invariants must include those that are of primary interest to users of such a system as well as those whose inclusion takes advantage of some of the more powerful extremal graph theory results.

The remainder of this paper is intended to be a brief informative discussion about the basic structure and use of the system. In the last section several figures are presented which were taken directly from a computer terminal during an actual session and, hopefully, they will help indicate the usefulness of the system.

DESCRIPTION OF THE SYSTEM

The current version of INGRID is implemented on a VAX 11/780 in the PASCAL programming language. It consists of some 20000 lines of source code, 3000-4000 of which are documentation, and the object code resides in approximately 500,000 bytes of main memory.

INGRID currently maintains the 37 graphical invariants listed in Appendix 2. Nine of these are binary valued, e.g., bipartite, planar, hamiltonian; one, spectral radius, is real valued; and the remainder are integer valued. For the most part binary invariants are included as a convenience for the user since their values can usually be determined directly from the numerical invariants, e.g., planar is true if and only if genus=0. Associated with each numeric invariant is an interval, or 'range', of one or more 'allowable' values and is defined by a 'lower bound' and an 'upper bound'. The initial lower and upper bound values of binary invariants are defined to be 'false' and 'true', respectively. INGRID assumes a graph has one or more connected

components with no isolated nodes and sets initial numeric lower bound values accordingly, e.g., nodes >= 2 and edges >= 1. Upper bound values for all numeric invariants are initialized to a large value which is set arbitrarily at 20000 and is referred to as 'infinity'. This value can be changed interactively by the user, but for larger values of infinity there is the possibility of causing arithmetic overflow problems.

The user of INGRID may, in a manner to be described below, restrict the current range of any invariant to any proper subrange. This may cause the range of other invariants to be narrowed by theorems which involve the restricted invariant. Such changes are recorded and they, in turn, may cause further repercussions. Eventually the system will 'stabilize' and await additional instructions.

Two points should be noted. First, the range of values associated with a particular invariant may, at any point in time, include invalid values for that invariant. In fact, it is possible that none of the values may be valid. A given range merely indicates that values not in the range are definitely invalid. The usefulness of a given range is limited by the completeness of the data base of theorems. Second, it is possible, during processing, that INGRID will determine for some invariant that its range must be empty. This constitutes an 'inconsistency error' and causes the system to automatically reset to the system status just prior to the user's last input. One of the causes of this 'error' can be an invalid theorem (more properly, a false conjecture). Another reason can be the incompleteness of the data base of theorems, as discussed above, and is not really an error. For example, if a user specifies a regular graph on 97 nodes INGRID will determine an 'allowable' range of values for the maximum (and minimum) degree to be [2,96]. The odd values in this interval are not specifically excluded at this point. However if the user then sets the maximum (or minimum) degree to, say, 13 an 'inconsistency error' will arise.

INGRID currently contains a data base of over 350 theorems. Each theorem is coded as an individual procedure which is invoked whenever a change has been made to any of the invariants of that theorem. For example, a well known result is that the diameter d of a graph can be no more than twice its radius r, i.e., d <= 2r. The procedure implementing this theorem does the following.

First,
it sets the upper bound of d to the minimum of the current upper bound of d and two times the current upper bound of r. Second, it sets the lower bound of r to the maximum of the current lower bound of r and the smallest integer greater than or equal to 1/2 of the current lower bound of d.

A user's initial input is viewed as a change in an invariant value
and sets in motion a chain of theorem, or procedure, executions. It should be clear that this process will eventually halt since the number of values in each invariant range is finite (even in the case of real valued invariants) and any change always yields a proper subrange of the original range. Therefore, the system must eventually 'stabilize' or encounter an inconsistency error. A compiled version of INGRID on a VAX 11/780 executes most commands
in 200-300 milliseconds of central processing unit time. Only a few exceptions have occurred when the upper bound on the number of
nodes was left unspecified and the particular specifications given
caused the lower bound to march slowly towards the upper bound of 'infinity'.

USING THE SYSTEM

The basic mode of user interaction with INGRID, initiated by a screen prompt of '>', is at the Command Level. All system messages
from INGRID to the user are preceded by '#'.

User commands are of two types:

(i) Invariant Setting(IS) commands and
(ii) System Action(SA) commands.

IS commands are the means by which a user specifies a value or range
of values for an invariant and are invoked by typing one of the valid
invariant names. SA commands allow the user to call upon INGRID to
manipulate the data bases in a variety of ways and generally enhance
the basic capabilities of the system. As with IS commands, SA commands are invoked by typing the name of the desired action. The exact format for both types of commands is presented below. In either case a truncated form of the name may be used by entering

only the first few characters--as many as necessary to form a stem
which uniquely distinguishes that name from any other invariant or
action name. Most IS commands and some SA commands require additional
input values. If this information is not given with the command name
(on the same line) the system will indicate the data needed and prompt
with '?'. When prompted, the user may enter the required information
or escape back to the Command Level by entering '#'.

The flow of control within INGRID is rather straightforward. The name
of the user specified invariant is placed onto a stack data structure.
This initiates a 'while' loop which executes as long as there are
elements in the stack. The while loop merely removes an invariant
name from the stack and invokes each theorem which uses that invariant.
The execution of a theorem will place other invariant names on the
stack if their range has been changed. Figure 1 presents this in
slightly more detail.

Suppose, for example, a user enters 'clique = 8'. This will cause
INGRID to invoke, among others, Theorem 220 which is

```
clique <= chromatic number(chr).
```

The lower bound of chr will be set to 8, and chr will be placed
on the stack. Processing will continue until, at some later
point in time, chr is removed from the stack. Now, all theorems
in which chr appears will be executed. Among these is Theorem 18,

```
chr <= (7+sqrt(1+48*genus))/2
```

which now sets the lower bound of genus to 2.

Invariant Setting Commands:

All invariant setting commands must adhere to one of the four
following formats. Syntactic units consist of an alphanumeric
'name', a numeric 'value', and special characters and may be
surrounded by any number of blanks.

(a) name=value or name value
(b) name:value,value or name:value value
(c) name>value or name>=value
(d) name<value or name<=value

Note: the syntactic unit '<' ('>') is interpreted as 'less (greater) than or equal'.

The semantics of the four forms are:

(a) the lower and upper bounds of the named invariant are set to value,
(b) the lower and upper bounds are set to the first and second values, respectively,
(c) the lower bound is set to value,
(d) the upper bound is set to value.

If the 'new range' is not a proper subrange of the old range, the system automatically returns to the Command Level with an appropriate comment and no changes are made to the invariant data.

A binary invariant command must use form (a) with a value field of 'y' (or missing) for true, and 'n' for false.

System Action Commands:

The general format of all System Action commands is:

name {list of zero or more parameters}

The specific name of each SA command and a brief description of its purpose is presented in Appendix 3.

A SESSION WITH INGRID:

This section describes a typical interaction with INGRID and shows the use of some of the IS and SA commands.

In Figure 2, four commands were issued. First, was the SA command
TIME which causes the generation, immediately following each IS command, of a system message which reports the time in milliseconds
and the total number of theorems executed. The next two were IS commands which set the number of nodes to 100 and limited the number of edges to less than or equal to 1000. The last,

an SA command of LIST, produced the display of invariants and their associated values or ranges. The pair of values within parentheses, next to each invariant, represents the current allowable range of that invariant. This is replaced by a single value if the range consists of just one element.The pair of values
in square brackets represents the 'previous' allowable range and is
present only when the current range is different from the previous
range. In Figure 2, the previous range is the initial range established by the system. In other instances the previous range will be the current range displayed by the previous execution of the SA command LIST. The symbol 'udt' implies that no value has been determined, i.e., it is essentially 'infinity'. The symbol 'undet' plays a similar role for binary invariants but here simply means the choice is, or was, undetermined.
A '*' before an invariant name is a reminder that the invariant has been the object of some previously issued IS command. The theorems which actually produced changes in the ranges of invariants are recorded in a 'rulestack' and may be listed, in the
order they occurred, by issuing the SA command RULES.

Figure 3 shows a continuation of the 'session' of Figure 2 with an
additional IS command restricting the maximum degree to no more than 30.

A more interesting use of INGRID is detailed in Figure 4. At a recent(1980) conference on graph theory[2], Professor Roger Cook posed the following problem.

> 'Does there exist a planar, triangle-free graph with a number of nodes equal to 3 times its node independence number?'

This problem can be presented to INGRID in any of several different forms. The following is an account of the approach that was actually used and has the additional benefit of exhibiting another feature of the system. This feature allows the user to interactively specify some numeric relationship that is to temporarily (until removed by the user) exist between two or more invariants without having to give each of them specific values. Such a relationship is decoded and executed in much the same way as any other relationship (or theorem) in the system and is therefore called a 'temporary theorem'.

First INGRID was reinitialized by issuing an END statement. Then a 'temporary theorem' was given which would force

the number of nodes to be equal to three times the node independence number (nind). Next, clique, the maximum clique size, was set to 2, which is equivalent to specifying that the graph is triangle-free. Finially, "nind" was arbitrarily set to 20 and the result listed. Note that INGRID states that any graph satisfing all these conditions must be nonplanar.

This, of course, does not prove that no such graph exists for values of "nind" other than 20, but upon examining the 'rulestack',
Figure 5, the determination of 'nonplanar' is seen to have been made by Theorem number 35 (item 106). All theorems in INGRID are listed in [1]. There, Theorem 35, which originally appeared in [3],
is given as

```
'If G is planar then either
      (i)  clique >= 3, or
      (ii) nind >= (nodes+1)/3'
```

and settles the question for all values of "nind".

CONCLUSION

It would appear, from this last example, that INGRID does not really prove theorems nor generate new information. On the other hand, whenever seemingly random data and disjoint facts are intelligently and systematically collected, organized and presented
there is, inherently, a gain in information. In this sense, INGRID
can generate new results and add to the base of knowledge in graph
theory. Other examples of this could be presented but they tend to
be more involved than the one given here. Often the conclusion is
not as obvious and is only detected by observing trends in certain
values as other values are systematically altered. Also, there is
usually more than just one theorem involved in obtaining a specific
result. The exact determination of the 'suggested theorem' is not
always straightforward. A major improvement in the system has been
the addition of a 'traceback' feature which automatically prints the
sequence of theorems leading to an observed result.

Other areas in which INGRID has been helpful include testing whether a 'new' theorem produces better bounds than theorems already in the system (by utilizing the temporary theorem feature) and, in a similar manner, testing conjectures. It also is capable of pointing out weaknesses in the data base of theorems and/or the graph theory literature itself and thus indicating possible directions for further research. Finially, working with INGRID can be educational for those that are just becoming familiar with graph theory in the sense that it can give them a 'feel' for how various invariants relate to each other.

APPENDIX 1

Basic Information

At the command level, the system prompts with ">".
A valid user response is any of the invariant names or commands. The value or range for invariants and any arguments for commands is expected on this same line.

If not given on the same line, the system will prompt with "?".

If a numeric invariant:
Enter either "= value"
": low high"
"< value", or
"> value".
(note: "<" and ">" are interpreted as "less than or equal" and "greater than or equal", respectively).

If a binary invariant: Enter "=y" or "=n".

When the given value or range does not reduce the current range the system will automatically return to the command level.

```
You may exit to the command level by entering  "!".

There are currently 383 theorems in the system,
  23 System Action Commands, and  37 invariants.
```

APPENDIX 2

Invariant Names and Meanings.

```
arbor  :  point arboricity number.
bwidth :  bandwidth.
circ   :  length of the longest cycle.
chr    :  chromatic number.
clique :  order of maximal clique.
diam   :  maximum of the maximum distances.
dom    :  domination number.
earbor :  edge arboricity number.
edges  :  number of edges.
eccov  :  minimum number of cliques to cover the edges.
echr   :  edge chromatic number.
econn  :  edge connectivity number.
ecov   :  minimum number of edges to cover the nodes.
eind   :  maximum number of independent edges.
genus  :  genus.
girth  :  length of the shortest cycle.
maxdeg :  maximum degree.
mindeg :  minimum degree.
nccov  :  minimum number of cliques to cover the nodes.
ncomp  :  number of components.
nconn  :  node connectivity number.
ncov   :  minimum number of nodes to cover the edges.
nind   :  maximum number of independent nodes.
nodes  :  number of nodes.
radius :  radius.
spectr :  spectral radius.
thick  :  thickness.
xnum   :  crossing number.

bipart :  bipartite graph      (binary, yes or no).
compl  :  complete graph                "
connct :  connected graph               "
cycle  :  cycle graph                   "
forest :  set of trees                  "
hamil  :  hamiltonian graph             "
planar :  planar graph                  "
reg    :  regular graph                 "
```

tree : connected,acyclic graph "

Names and commands may be abbreviated by truncating to a unique set of characters. For example the number of nodes may be specified by using any of:

nodes node nod no

APPENDIX 3

System Action Commands.

batch : In the default interactive mode long responses are broken up into screen size pages by the insertion of "pauses" which require the viewer to hit the return key inorder to proceed to the next page. These pauses are undesirable when output is routed to other than the screen. Issuing "batch" disables the pauses. To restore the pauses, simply issue batch again.

Examples: batch bat ba

bstep : Backward step. Displays up to 9 Tables, in reverse order of occurrance. After each is displayed, the viewer may continue or, by typing "!", end the command.

Examples: bstep bst bs

bound : With invariant name argument(s) i1 i2 ..... The interval for each of i1 i2 ... is printed. Usually used when a value is too large to be printed in the normal Table display.

Examples: bound nodes chr t bo sprectr

date : Displays the current date and time.

Examples: date da

dtt : Deletes the last temporary theorem. With argument i, deletes the i\th temporary theorem. With argument a, all temporary theorems are deleted.

Examples: dtt dtt3 dtt 10 dtt a

end : Terminates the current graph.

Examples: end en

exclude : With argument i, causes permanent theorem number i to be deactivated. With argument tt i, temporary theorem i is deactivated.

Examples: exclud 54 ex 10 40 3 ex tt 4

fstep : Forward step. With argument i, tables i, i+1,... are displayed. With no argument, the lowest possible value is used.

Examples: fstep 4 fs 3 fs

ftrace : Full text trace. See "trace".

ftwith : Full text of theorems with i1, i2, ... See "twith"

help : Provides review of basic information.

Examples: help he

include : With argument i1 i2 ... causes theorems i1 i2 ... to be reactivated (after they have been "excluded"). An argument may also be tti (or tta) to reactivate a (or all) temporary theorems.

Examples: include 5 30 19 in 17 tt3 29

list : Displays the current table of invariant values. With argument a, invariants are presented in alphabetic order.

Examples: list li li a

recall : With argument i, displays the i\th data base.

Examples: recall 4 rec 6

remove : With argument i, deletes the last i tables.

Examples: remove 3 rem 6

rules : Prints a list, in chronological order, of theorem numbers. The theorem corresponding to a number was invoked and altered the table which was current at that time.

Examples: rules ru

thmtext : With arguments i1 i2 ... , prints the text of the theorems numbered i1 i2 ... . Any argument may be tti (or tta) and prints the (or all) temporary theorem(s) i.

Examples: thmtext 3 56 thm 29 tt3 97

time : Activates(and deactivates) an instruction timing device. The default is no timming. When activated the time in milliseconds of each command issued is displayed after its execution.

Examples: time ti

trace : With argument i> (or i<) presents the sequence of rules resulting in setting invariant i"s lower(or upper) bound. A one line desciption of each rule is printed. You may use "ftrace" to obtain a more complete description.

Examples: trace nind< tra edges>
ftrace conn y ft conn <

tt : Accepts a temporary theorem for inclusion in the knowledge base. The temporary theorem remains until either deleted by the command dtt or the session ends.

The user supplied theorem must be of the following form:

"invariant name" RELOP "arithmetic expression"

where RELOP may be any of
"<" (or "<=", these are treated the same here)
">" (or ">=", "
"=" (or blank, "

The arithmetic expression must be in "standard" infix form with the following restrictions: an invariant name can occur only once and must be different from that appearing before RELOP; the final value of the expression must be monotonic in each invariant; and the allowable operators are +(add), -(subtract), /(integer divide), and *(multiply).

Examples: tt chr <= maxdeg/2
tt nconn econn
tt gi < ma+min

twith : With argument(s) i1 i2 ..., prints the identification numbers of theorems which contain the invariants i1, i2 ... . May use "ftwith" to obtain the numbers and the complete text of such theorems.

Examples: twith edges maxdeg girth
tw chr ma nconn
ftwith ma mi chr
ft reg ha

ucomm : prints the sequence of user given commands (those altering the table.

Examples: ucomm uc

undo : Removes the last command and its effects. Note that undo followed by undo does nothing.

Examples: undo un

versus : With argument i, causes the current data base to be compared with the i\th(the given argument) and listed.

Examples: versus 3 ve 4

Names and commands may be abbreviated by truncating to a unique set of characters. For example the command versus may be specified by using any of:

versus versu vers ver ve v

APPENDIX 4

Adding a New Command

A new command, say, "newcom" is added to the system as follows.

(a) In the "const" section (approx. line # 164) decrement the constant "ninst" by one (note that it is a negative integer) to "-24". Then add a new line of

```
newcom:=-24;
```

(b) In "procedure startup", at approx. line # 485, add

```
parameter[newcom]:='newcom';
```

(c) Add a procedure, say "pnew", to execute the new command (anywhere prior to "procedure input".)

(d) In "procedure input", add a new case to the case statement - "case x of .... end".
The new case should be

```
new: pnew;
```

APPENDIX 5

Adding a New Theorem

Each theorem "iii" requires a procedure named "riii". The constant "cnumrules" (found in the "const" section) is the number of theorems the system is designed to hold. The last few of these may be just empty shells. The variable "actualnumrules" (found in procedure "startup") is the number of theorems actually implemented at this time. If there are no empty shells (actualnumrules=cnumrules) you will have to create one in step (a) below. The number assigned to your new theorem will be nnn (=actualnumrules+1).

At this time cnumrules= 460 and actualnumrules= 458, so you will not have to add an empty shell in (a) below.

(a) First, if necessary, increase "cnumrules" by one. It is in the "const" section. Also, if necessary, you must add a new empty shell for the procedure for your new rule (right before "procedure delthm"). The form of the new empty shell is:

```
procedure r459;
(*************************)
(*                       *)
(*   documentation       *)
(*                       *)
```

```
(*************************)
begin
  if activerule[459] then
    begin
      rule:='359/ ';
        .
       code
        .
    end;
 end;
```

(b) In "procedure executerule(i:integer);", add a new case to the case statement:

```
        459: r459;
```

(c) In "procedure eval;", do the following:
if your new theorem uses(in procedure r459) invariants x, y, etc., add to each case x, case y, etc. a statement - "r459;"

(d) In "procedure ruletext", near the end, add a write statement to print the text of the new theorem. Note if the description only reguires one line then

```
     459: write(' .....  ');
```

is sufficient. Otherwise you must check to see if a "short" version is wanted (trace = 1) or a longer more complete version is desired.

(e) In "procedure startup", find the set assignment statements- they are of the form (for theorem iii)

```
       rulemx[459]:=[a,b,...];
         rulemn[459]:=[x,y,...];
```

The MAX of invariants a, b,... and the MIN of invariants x, y, ... are used in the evaluation of theorem iii (i.e., in "procedure r459;").

You must add similar statements for the new "459".

At the end of this group of statements (probably where you just added) is the assignment for the variable "actualnumrules". It must now be assigned 459.

Also, just following this, and if the text description given in step (d) requires k > 1 lines, then add the statement

```
         lsinthm[459]:=k;
```

APPENDIX 6

Adding a New Invariant

To add a new invariant name, say xxxxxx. The name must be 1-6 alphanumeric characters with no embedded blanks and, if it is k characters in length, they must be different from the first k characters of any othe name (invariant or command).

(a) In the "const" section, increase the value of the constant "nparam" by one to 37. Then add a new constant with value nnn as follows:

```
xxxxxx= 37;
```

(b) In "procedure startup", add the statement

```
parameter[xxxxxx]:='xxxxxx';
```

If xxxxxx is to be a binary invariant, add its name to the binary set assignment statement:

```
bparam:=[xxxxxx,.....];
```

(c) In "procedure eval", add a new case to the case statement:

```
xxxxxx: begin
          riii;.....
        end;
```

where the listed pocedure calls riii, etc., are (or will be) theorems which use invariant xxxxxx

REFERENCES

1. R. C. Brigham and R. D. Dutton, A compilation of relations between graph invariants, Department of Computer Science technical report TR-59, University of Central Florida, March 1983.

2. R. J. Cook, in The Theory and Applications of Graphs(Proc. of the Fourth International Conf. on the Theory and Applications of Graphs, May 6-9, 1980, Western Michigan Univ., Kalamazoo, Mich.), Edited by G. Chartrand, Y. Alavi, D. L. Goldsmith,

L. Lesniak-Foster and D. R. Lick, John Wiley and Sons, New York,
1981, pg. 610.

3. K. Walker, The analogue of Ramsey numbers for planar graphs, Bull. Lond. Math. Soc. 1 (1969), pgs. 187-190.

€
ands of up to 37 graph invariants. 5he remaining graph invariants. start over from the beginning. Facilities also exist which aid a INTRODUCTION
G T are perhaps s , INGRID has also been transported to the Macintosh. . T , and. T that,,to A that , ,occur4 enthat corresponding
zT zU zV zW zY zZ z]
z^ za zb z z, z¡ zÀ zß zà zá zâ zî zï zð zñ zò zó zô zõ zö z÷ zø zù zú zû { {
{
{

{
{□ {□ {□ {□ {□ {□ {□ {□ {□ {□ {□ {□ {□ {□ { {
{ { {R {S {V {W {X {Y {Z {[{` {c {d {e {f {g
{h {i {j {k {l {m {n {o {p {q {r {s {t {x {y {z {
| {}
{~ { {€ { {, {ƒýú ú ýôññýýýýýýýýýð ýýýýýýýýýýýýýýýýýýýýýý
ýý □€□€□ □
@ □ □ □ □ □ □ □ □_ □ □□ □ □A _□B □n □o □p □q □…
□À □Á □Þ □ß W X „ ® ¯ °
± Ð Ñ □õ □ö □÷ □□ □□ □C □… □Ç

M • Ò □ □ ` □ ¦ □ é + o ° ò

7
}
Ä
ã
ä
'
h
©
ê

+

,

-

.

H

I

Š

Í

O
P

Ò O ’ Ð Y • Ï S • Ø [ž Ý
I ‹ Æ I “ ”
Øõõõõõõõõõõõõõõõõõõõõõõõõõõõõõêõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõ
õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõ

!À
´ P!¿
!À
´ P!¿ \ Ø Z œ â ' j °
ï 2 t ¹ ÿ @ † É

□S □„ □… □Ã □□ □H □‹ □Ï □□ □V □—
□Ù □□ □< □= □□ □Ä □ F ^ É

Q “ × ^ j k l m ~ ¿ 1 2 T
U ‹ º » H Ó ! !T !– !
Û " "c "« "ò #: # #¸ #¹ $ $G $Š $Î % %\ %Ÿ %
µ %¶ %
ù &. &= &i &j &« &é '+ 'n 'p '™ 'š 'È 'Éõõõõõõõõõõõõ
õõõ
õõõõõõõõõõõõõõõõõõõõõ
!À
´ P!¿ a 'É 'Ê 'Ë 'ç 'è (' (g (¥ (É (Ê (ý)6)
j)ž)Ÿ)Û)õ)ö * * *_ *
€ *À *ñ +" +R +S +’ +Ñ , , ,V ,‡ ,ˆ ,‰ ,Š ,¢ ,
£ ,Ø ,Ù - -
-H -s -t -u -v - -Ž -Ì -ý -þ .B .ƒ .È /
/E /„ /Ã 0 0? 0€ 0Ä 1 1M 1‹ 1È 2 2@ 2
| 2½ 3 3F 3† 3Á 4 4; 4< 4€ 4Á 4Ê 4Ë 5
5I 5k 5l 5-
5é 5ë 5ì 6' 6e 6Ÿ 6ß 7 7V 7’ 7Ðõõõõõõõõõõõõõõõõõõõõõõõõ
õõ
õõõõõõõõõõõ
!À
´ P!¿ a 7Ð 8
8T 8U 8“ 8Ë 9 9B 9, 9Á 9ý :9 :: :y :¾ :ÿ ;B ;
ˆ ;” ;• ;¼ ;â <
< <A <B <C <N <O <Ž <Ñ = =S =˜ =Ü >! >e >© >
í ?2 ?v ?º @ @6 @7 @8 @~ @Ã A
AL A AÔ B Ba B© Bª B« B¬ B B® B¯ B°
BÎ BÏ BÐ Bñ Bò C% CY C CÆ CÇ Cý D D D# D?
D^ D Dœ DÞ E! EG EH Ew Ex Ey E³ Eô Eõ F)
F* FZ F‹ FŒ F FŽ F õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõ
õõõ
!À
´ P!¿ a F F F¬ F-
F® FÖ F× Fú G G6 GR Gu G¡ G¾ Gà Gû H3 HT Hx H®
HÝ Hî I I0 IJ I, I¢ IÆ Iü J+ JF JX Js Jˆ J£ J
¤ JØ K K, KV K€ Kª KÔ Kþ L(L)
L* L+ L, Lc L• LÊ LË Lê Lë Lì Lí Lî Lï Lð M M
M M: M; M< M{ M¹ Mü N>
N~ N¾ Nø Nù O1 O2 Og O¬ Oã P P P6 P7 Pt P³ Pó
Q Q QU QV Q† Q‡ Q°
Q± Qò R3 Rc Rdõõ
õõ
!À
´ P!¿ a Rd R¦ R§ RÑ RÒ Rû Rü S?
S SŸ S Sá Sâ T# To Tp T¢ T
£ TÍ TÎ TÏ U U U UD UE Uo Up U® Uï V. VY VZ Vš
V›
V× W W. W/ Wb Wc W Wž WÊ WË X X X4 X5 Xs X²
Xñ Y Y Y5 Y6 Yt Y³ Yï Z Z
ZD ZE Zƒ ZÁ [[+ [, [Z [[[š [Ø \ \S \u \v
\¬ \á \â]]_]¢]á]ó ^3 ^4 ^Z ^ ^Î ^ÿ _A _
€ _À `  `@ `}  `-
`®õõ

õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõ
 !À
 ´ P!¿ a `® `Û a a-
a. ap a± aò b" b# bU b~ b¨ bË bÌ c
c* c+ cY cZ cš cÊ cË cö c÷ d7 dz d{ d© dª d« dç
e eQ eR eƒ e„ e… e† e‡ eˆ e¦ e§ e¨ eÌ eÍ f f
fM f f¿ fÀ fÞ fß g g gI gJ gŠ g¹ gº gó h h< h
= hZ h[h\ h] h^ h_ h` ha h} h~ h h¡ h¢
hÝ i iT iˆ i¿ iø j6 js j
¥ j¦ jÝ k k kS k’ kÎ l
l(l)
l*õõõ
õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõ
 !À
 ´ P!¿ a l* lB lg lŒ l± lÖ lû m
m+ m= mZ ml m€ m’ m£ m± m² m³ mñ n n)
n* nY n‘ nÐ nú nû o5 os o« o¬ oÒ oÓ p pP pv pw
p® pè pé q q9 q: qw q´ qî qï r* r+ rd r
¥ rÜ rÝ s sY sk sl sŽ s s s‘ s’ s°
s± s² sÙ sÚ t tZ t˜ tÌ tÍ u u= uh ui u„ u… u¶
u· uâ uã v vM vN vx vy v° vÂ vÃ và w w w!
w` w w¡ w¢
õõ
õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõ
 !À
 ´ P!¿ a w¢ w£ w® w¯ wï x-
xi xy xz x½ y yB y, yÈ yÛ yÜ z zR zS zT zU zV
zW zX zY zZ zï {ó : Œ – › —
ŠõõõõõõõõõõõõõõõõõõõõõêåêååõÚÏÏÏÏÏÏå

 !À
ýÒ ´ #(
 !À ýÒ ´ #(
 !À ´ P!¿
 !À
 ´ P!¿ !
  Þ

.Theorem 220

are

e called a 'temporary theorem'.

,

System Action Commands.

ton, A compilation of relations Department of Computer Science
Applications of Graphs(Proc. of tern Michigan Univ., Kalamazoo,
Orlando, Florida 32816

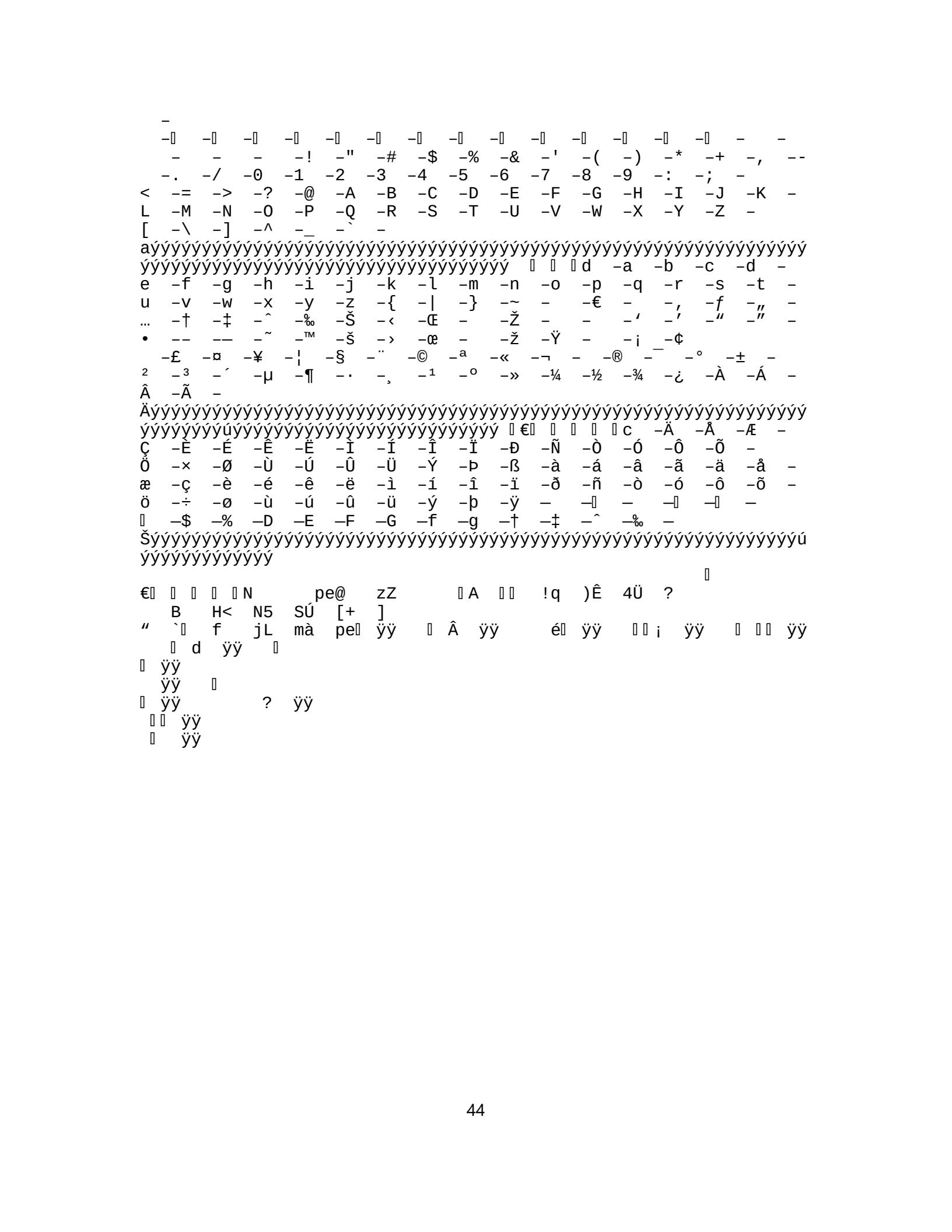
ABSTRACT

–f pNu < 2£ f Rp Cúÿj"ˆ1x
Z 1| ,Nup a {ƒ {„ {… {‰
{Š {‹ {Œ { {Ž { { {' {' {" {" {• {– {—
{˜ {™ {š {› {œ { {ž {Ÿ {_ {¡ {¢
{£ {¤ {¥ {¦ {§ {¨ {© {ª {¯ {° {± {³ {·
{¸ {Å {Æ {Ç {È {É {Ê {Ë {Ì {Í {Î {Ï {Ð {Ñ {Ò {Ó
{Ô {Õ {Ö {× {Ø {Ù {Ú {Û {Ü {Ý {Þ {ß {à {á {â {ã
{ä {å {æ {ç {è {é {ê {ë {ì {í {î {ï {ð {ñ {ò {ó {
ô {õ {ö {÷ {ø {ù {ú {û {ü {ý {þýýýýýýýýýýýýýýýýýýýýýýýýýý
ýýý
ýýýýýýýýýýýý d {þ {ÿ |
! " # $ % & ' ( )
4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R X Y Z [\ ]
^ _ ` a b c d e f g h i j k q r
s t u v w x y z { | , ƒ „ ‡ ˆ
‰
Š ‹ Œ Ž ýý÷ýý
ýý
@ b ' ' " " • – —
˜ ™ š › œ ž Ÿ ¡ ¢ £ ¤
¥ ¦ § ¨ © ª « ¬ ® ¯ °
± ² ³ ´ µ ¶ · ¹ º »
¼ ½ ¾ ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë
Ì Í Î Ï Ð ï ð ñ ò ó ô õ ö ÷ ø ù
ú û ü ý þ ÿ ' ' ' ' ' ' ' ' ' '
'
'
'

“
“□ “□ “□ “□ “□ “□ “□ “□ “□ ýýýýýýýýýýýýýýýýýýýýýýýýýýýýýýýý
ýý
ýýýýýý □ □ □d “□ “□ “□ “□ “□ “□ “ “ “ “ “ “!
“" “# “$ “% “& “' “(“) “* “+ “, “-
“. “/ “0 “1 “2 “3 “4 “5 “6 “7 “8 “9 “: “; “< “=
“> “?
“@ “A “B “C “D “E “F “G “H “I “J “K “L “M “N “O
“P “Q “R “S “T “U “V “W “X “Y “Z “[“\ “]
“^ “_ “` “a “b “c “d “e “f “g “h “i “j “k “l “m
“n “o “p “q “r “s “t “u “v “w “x “y “zýýýýýýýýýýýýýýý
ýý
ýýýýýýýýýýýýýýýýýýýýýýý □ □ □d “z “{ “| “}
“~ “ “€ “ “‚ “ƒ “„ “… “† “‡ “ˆ “‰
“Š “‹ “Œ “ “Ž “ “ “‘ “’ ““ “” “• “– “—
“˜ “™ “š “› “œ “ “ž “Ÿ “ “¡ “¢
“£ “¤ “¥ “¦ “§ “¨ “© “ª “« “¬ “ “® “¯ “°
“± “² “³ “´ “µ “¶ “· “ “¹ “º “»
“¼ “½ “¾ “¿ “À “Á “Â “Ã “Ä “Å “Æ “Ç “È “É “Ê “Ë
“Ì “Í “Î “Ï “Ð “Ñ “Ò “Ó “Ô “Õ “Ö “× “Ø “Ù “Ú “Û
“Ü “Ý “Þýýý
ýý □ □ □d “Þ “ß “à
“á “â “ã “ä “å “æ “ç “è “é “ê “ë “ì “í “î “ï “ð
“ñ “ò “ó “ô “õ “ö “÷ “ø “ù “ú “û “ü “ý “þ “ÿ ”
”□ ” ”□ ”□ ”□ ”□ ” ”□ ” ”
”
”

″
″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″▯ ″ ″
″ ″ ″ ″! ″" ″# ″$ ″% ″& ″' ″(″) ″* ″+ ″, ″-
″. ″/ ″0 ″1 ″2 ″3 ″4 ″5 ″6 ″7 ″8 ″9 ″: ″; ″
< ″= ″> ″? ″@ ″A ″
Býýý
ýýý ▯ ▯ ▯d ″B ″C ″D ″E ″
F ″G ″H ″I ″J ″K ″L ″M ″N ″O ″P ″Q ″R ″S ″T ″U ″
V ″W ″X ″Y ″Z ″[″\ ″] ″^ ″_ ″` ″a ″b ″c ″d ″e ″
f ″g ″h ″i ″j ″k ″l ″m ″n ″o ″p ″q ″r ″s ″t ″u ″
v ″w ″x ″y ″z ″{ ″| ″} ″~ ″ ″€ ″ ″‚ ″ƒ ″„ ″… ″
† ″‡ ″ˆ ″‰ ″Š ″‹ ″Œ ″ ″Ž ″ ″ ″‘ ″’ ″“ ″” ″• ″–
″— ″˜ ″™ ″š ″› ″œ ″ ″ž ″Ÿ ″ ″¡ ″¢
″£ ″¤ ″¥ ″¦ýýý
ýýý ▯ ▯ ▯d ″¦
″§ ″¨ ″© ″ª ″« ″¬ ″ ″® ″¯ ″° ″± ″² ″³ ″´ ″µ ″¶ ″·
″¸ ″¹ ″º ″» ″¼ ″½ ″¾ ″¿ ″À ″Á ″Â ″Ã ″Ä ″Å ″Æ ″
Ç ″È ″É ″Ê ″Ë ″Ì ″Í ″Î ″Ï ″Ð ″Ñ ″Ò ″Ó ″Ô ″Õ ″
Ö ″× ″Ø ″Ù ″Ú ″Û ″Ü ″Ý ″Þ ″ß ″à ″á ″â ″ã ″ä ″å ″
æ ″ç ″è ″é ″ê ″ë ″ì ″í ″î ″ï ″ð ″ñ ″ò ″ó ″ô ″õ ″
ö ″÷ ″ø ″ù ″ú ″û ″ü ″ý ″þ ″
ÿ • •▯ • •▯ •▯ •▯ •▯ • •▯ • •
ýý
ýýý ▯ ▯ ▯d •
•
•

•
•▯ •▯ •▯ •▯ •▯ •▯ •▯ •▯ •▯ •▯ •▯ •▯ •▯ •▯ • •
• • • •! •" •# •$ •% •& •' •(•) •* •+ •, •-
•. •/ •0 •1 •2 •3 •4 •5 •6 •7 •8 •9 •: •; •< •=
•> •?
•@ •A •B •C •D •E •F •G •H •I •J •K •L •M •N •O
•P •Q •R •S •T •U •V •W •X •Y •Z •[•\ •]
•^ •_ •` •a •b •c •d •e •f •g •h •i •j •k •l •mý
ýýýúýýýýýýýýýýýýýýýýýý
ýýýýýýýýýýýýýýýýýýýýýýýýýýýýýýýýýýý ▯
€▯ ▯ ▯ ▯ ▯c •m •n •o •p •q •r •s •t •u •v •w •x •y
•z •{ •| •} •~ • •€ • •‚ •ƒ •„ •… •† •‡ •ˆ •‰
•Š •‹ •Œ • •Ž • • •‘ •’ •“ •” •• •– •—
•˜ •™ •š •› •œ • •ž •´ •µ •¶ •·
•¸ •¹ •º •¾ •¿ •À •Á •Â •Ã •Ä •Å •Æ •Ç •È •É •Ê
•Ë •Ì •Í •Î •Ï •Ð •æ •ç •è •é •ê •ë •ì •í •î •ï
•ð •ñ •ò •ó •ô •õ •ö •÷ •ø •ù •ú •û •ü •ýýýúýýýýýýýý
ýý
ýýýýýýýýýýýýýýýýýýýýýýýý ▯€▯ ▯ ▯ ▯ ▯c •ý •þ •ÿ – –▯ – –
▯ –▯ –▯ –▯ – –▯ – –
–
–



 ÿÿ

 ÿÿ 9 ÿÿ
 ÿÿ
P ÿÿ > ? Z [\ ] ”
  • Ë æ ç   ) * < = >
 K L
B
C
D
^
_ * + ¯ °
  † ‡ / 0 M N O `  a  2  ` ^ ‰ !p !
q #d #e #ã $
 $
 % %
 %/ %0 %^ %_ %` %| %} &Z &[&‰
 &º &é ' ' 'm 'n '’ '” 'Û (1 (W (| (})=)>)°
)±)²)Ê)Ë * *. */ *˜ *™ *š *± *² +! +" 1> 1?
_ 1Ë 1Ì 2i 2j 2â 2ä 5@ 5A 7 7 8o 8p 8’ 8
 8Ò 8Ó 9 9 9 9 9 <² <³ ? ?(?) ?* ?< ?
= @ @ @R @S @l @† @œ @´ @È @ÿ A7 AR AS A~ A B-
 B. BŒ B˜ B™ Bš B¸ B¹ BÚ Bî C C, CN Cx C’ C³ CÌ
 D D! DC Dw D¤ D³ DÙ Dò E
 EA E_ E Eµ Eâ Eû F

F& F9 FR FS Fƒ F
£ FÅ Fä G G& GF Ge GŠ G‹ GŒ H" H# H; HG HH HI Ha
Hb HŸ H× I IP IŠ IÄ Iø Iù J!
J" JT J“ JÄ JÛ JÜ K K K= Kv K°
KÌ KÍ L L L. L/ LM LN LŒ LÇ Lñ Lò M# M
$ MK ML Mi Mj M« Mç Mÿ N N4 N5 Nt N° N»
Nß Nà O O O
OI OJ OK Oz O{ O™ Oš O× P PK Pp Pq P¤ P
¥ PÞ Q Q) Q* QO QP Q^ Q‰ Q« Q¬ Qâ Qã R R
RE R~ R·
RÈ RÉ Ré Rê S' S_ S• S© Sª SÚ SÛ T TN T‡ T¬ T-
TÍ TÎ U
UC U} U² UÎ UÏ Uû V" V# V\ V– VÓ W

W WP WQ Wq Wª WÊ Wë X' X` Xš XÔ Y YE Yo Yp Y•
Y± YÍ YÎ Z ZI Zƒ Z-
Z® ZÙ Zö [[+ [, [h [, [ƒ [£ [¤ [á \
\

\+ \, \j \§ \¨ \Ë \Ì \Í]▯]7]k]l]’]ž]
Ÿ]]µ]¶]÷]ø ^
4 ^p ^ž ^Ÿ ^µ ^¶ ^ï ^ð _▯ _▯ _U _
€ _ _› _Þ _ù _ú `▯ `▯ `▯ ` `1 `2 b4 b5 b£ b
¤ bß c▯ cT c c¨ c© cª c¾ cß d
d* dU dv d
€ d d© d¿ dÉ dÖ dà dë dõ dö d÷ e4 eO eb ec e‘ eÃ
eü f f fX f’ fÆ fÇ få fæ g g[g}
g~ g´ gë gì h

h- h. hd h› hÏ hÐ i i i= iz i-
i® ié j" j0 j1 jK jW jX jY jp jq ka kb kš kÍ kõ
kö l
l
l; l< l` la l™ lÅ lÆ lç lè m m- m. mD m
\ mm mn m¦ mß më mì n¦ n§ oî oï p^ pe !À
!À
!À !À
!À !À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À

!À
!À
!À
!À !À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À

!À
!À
!À
!À
!À

 !À
 !À
 !À
 !À
 !À

!À
▯ !À
▯ !À
▯ !À
!À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
!À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
!À
▯ !À
▯ !À
▯ !À
▯ !À
!À
▯ !À
▯ !À
▯ !À
!À

▯ !À
!À
▯ !À
!À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À

!À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
!À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À

▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
!À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À

▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À
▯ ! À

▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À

▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À

▯ !À
    ▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
    ▯ !À
▯ !À
▯ !À
▯ !À
!À
▯ !À
!À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À

▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
    ▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À
▯ !À

!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À
!À

{ƒ {þ ‘ ‘r ‘Õ ’N ’2 “ “z “Þ ”B ”¦ •
•m •ý –a –Ä —Š > L M N O P Q R S T U V W X Y Z [\ ]
Ø 'É 7Ð F Rd `® l* w¢ —Š ?
@ A B C D E F G

□ □ □ □ □ □ □ □ ! $ (- / 1 2 3 ? C L N c k n
€ ‚ ƒ „ … ‡ ˆ ‰ Š ‹ Œ – › ª ¶ É Ê Ë Ì Õ Ú å î ò ô□ Ð Ñ Ò Ó Ô
Õ Ö × Ø Ú Û Ó □ H H ó ?ÿîÿî□□ R□G□(□ü H H ó ?
□ d □ □□□ □'□ □ □ h□ □□□
=à/Ð□ □ □ □p ÐR □ □ □B H □□-:LaserWriter 8.0
M Chicago □New
York □□Geneva □□Monaco □□Venice □□London
□Athens □□Fletcher □Tiny
□Future
□Cairo

Los Angeles

Zapf Dingbats Palatino Lineal Times
Helvetica
Courier Symbol Premiere Uncial Silicon Valley !
Pierce $ Star Trek (pica -Obliqe /Dots 1
Detroit 2 Ravenna 3 Hamburg ?Padova C
imageWRITER LStiletto N

Santa Monica c Hanford k Elite n Marie
€ Finney Jones , OMEGA ƒ Palo Alto „ Mobile …
San Francisco ‡ Seattle ˆ Spokane ‰ Stencil Š Tiffany ‹
Vancouver Œ
Wall Street – LED › EPSILON LAMBDA ª
KAPPA BOLD ¶ L.A. É Infocom Ê
Park Avenue Ë NY Nights Ì Scan Õ

Square Serif Ú
Hood River å Dover î McCloud ò Storybook ô TrekFont
Rangers Đ

Arial Narrow Ñ Arial Ò

Book Antiqua Ó Bookman Old Style Ô Century Gothic Õ
Courier New Ö Monotype Corsiva × Monotype Sorts Ø Century
Schoolbook Ú Times New Roman Û Wingdings Ó MT
ExtraÀ > > € € > (Õ ü (
ý0 P!¿$ < J O ý0 P!¿$ < J O ý0 P!¿$ <
ý0 P!¿$ #(ý0 P!¿$ #(<
P!¿$ #(=
ý0 þü P!¿$ #(=
ý0 8 P!¿$ #(=
<
P!¿$ #(Ð P
!¿#($ ý0 P!¿#($ ý0 P!¿$
#(ý0 l þ˜ P!¿$ #(ý0 l þ˜ P!¿#($
ý0 l þ˜ P!¿$,< = > ?
Y Z] n o Ž “ • Ë ç) €
i ˆ ý ; c d l m € > J K ˆ ‰
É Ê ä å
M N m n w x | }
~ Œ À Á Ò Ó ß $ %
H I ‹ Œ Ð Ñ æ ç ñ õ ö Q R ”
• ¬ Ó Ô X Y ž ã ä E F
, „ … • – Å Æ
B
D
^
ž
Ÿ
¿
À
ó
G
J
L
M
Í
Î
Ü
Ý

"

'

L

0

o

p

^

‰

Ê

Ë

@
A
K
L

Ž
Ç
È
Ý
Þ
ÿ A B O P
€ Á Â ü ý E F L P W X d e f
 h „ … ‘ Å Æ + , l m § ¨ ¬
  æ ç ð ô #
$ d e Ë Ì ñ ò 0 1 S T p q ‘ ’ °
 ± å æ ô õ 8 9 B C y z ¾ ¿ ü ý >
 ? J K ~ « ¬ Á Â
 " # E F Š ‹ Ç È Ë Ì
  S T _ ` „ µ ¶ Â Ã Ó
 Õ á ë ï ú û 7 D E M N … † Ç È

 K L , ƒ Š ‹ Ë Ì

□
□r □s □³ □´ □ó □ô □3 □4 □t □u □□ □ž □³ □´ □ô □õ
□8 □9 □q □r □x □y □»
□¼ □þ □ÿ @ A M O ` Ÿ Î Ï â ã □ □
0 1 2 3 4 5 6 7 8 ; < = ` a b c
d e f j k ‡ ˆ ‰ Í Î □ □ $ %
V W ›
œ Á Â Ù Ú □ □ X Y œ ß à " #
/ 0 h i v w ® ô õ þ ÿ !9 !: !n !o !
¶ !· !ý !þ "? "@ "V "W " ", "É "Ê #

```
#
#3  #4  #M  #N  #–  #—
#¦  #§  #Ò  #Ó  #Ô  #ß  #â  #ã  #ä  #å  #æ  #ç  #è  $
$
$J  $K  $X  $Y  $†  $‡  $´  $µ  $Æ  $Ç  $í  $î  %
%
```

```
  %
  %□  %□  %□  %^  %`  %|  %º  %»  %Í  %Î  %ø  %ù  &5  &6  &[  &
\  &]  &^  &_  &`  &j  &o  &p  &q  &r  &s  &y  &z  &|  &}
  &ƒ  &ˆ  &‰  &Š  &‹  &Œ  &  &Ž  &˜  &£  &¤  &
¥  &§  &¨  &®  &¹  &º  &»
  &¼  &½  &¾  &¿  &É  &Î  &Ï  &Ð  &Ñ  &Ò  &Ø  &Ù  &Û  &Ü  &ã  &è
 &é  &ê  &ë  &ì  &í  &î  &ø  &ý  &þ  &ÿ  '  '□  '□  '  '  '
  '□  '□  'R  'S  'Ž  '‘  '’  '“  'È  'É  'Ô  'Ù  'Ú  'Û  (
       (□  (  (1  (O  (U  (W  (u  (z  (|  (º  (»  (ø  (ù  );  )
<  )=  )>  )}  )~  )®  )¯  )°  )±  )²  )É  )
ÿ  *  *□  *  *□  *□  *□  *,  *-  *.  */  *k  *l  *–  *—
  *š  *±  *î  *ï  +  +  +!  +"  +I  +J  +c  +d  +£  +
¤  +ç  +è  ,□  ,□  ,$  ,%  ,a  ,b  ,”  ,•  ,ž  ,Ÿ  ,Ü  ,Ý  ,ÿ  -
  -□  -□  -U  -V  -x  -y  -•  --  -Ø  -Ù  .  .  ._  .
`  .f  .g  .›  .œ  .¶  .·  .Ö  .
×  /□  /□  /2  /3  /K  /L  /†  /‡  /’  /“  /Å  /Æ  0
  0
```

0
0L 0M 0† 0‡ 0Š 0‹ 0Ä 0Å 1
1 1 1, 1Á 1Â 2 2 2 2 2F 2G 2j 2§ 2
¨ 2á 2â 2ã 2ä 3 3 3Z 3[3“ 3” 3½ 3¾ 3Ñ 3Ò 4
4 4F 4G 4 4, 4š 4› 4½ 4¾ 4ù 4ú 5 5? 5| 5}
5³ 5´ 5ë 5ì 6(6) 6g 6h 6 6€ 6¤ 6
¥ 6Á 6Â 6Þ 6ß 7Y 7Z 7 7ž 7Ý 7Þ 8 8 8 8 8-
8. 8b 8c 8p 8q 8r 8s 8t 8u 8w 8’ 8“ 8” 8• 8– 8—
8˜ 8™ 8š 8ž 8Ÿ 8¯ 8° 8± 8² 8³ 8´ 8µ 8¶ 8·
8Ñ 9 9 9 9# 9$ 9P 9Q 9] 9^ 9™ 9š 9Ý 9Þ :
:
: :! :c :d :k :l :• :– :§ :¨ :Ñ :Ò :é :ê ;, ;-
;_ ;` ;n ;o ;² ;³ ;º ;» ;ô ;õ <

<
<7 <8 <| <} <°
<± <² <÷ <ø =; =< = =, =ž =Ÿ =Á =Â >
H >Œ > >Ó >Ô ? ? ?' ?(?* ?< ?o ?p ?£ ?¤ ?Ú ?
Û @G @H @S @T @l @m @n @† @‡ @ˆ @‰
@Š @‹ @Œ @ @Ž @ @œ @ @ž @Ÿ @ @¡ @¢ @£ @¤ @
¥ @´ @µ @¶ @· @¸ @¹ @º @»
@¼ @½ @È @É @Ê @Ë @Ì @Í @Î @Ï @Ð @Ñ @Ò @Ó @ÿ A
A A A A A A A A A
A7 A8 A9 A: A; A< A= A> A?
A@ AA AB AS Ai Aj A~ A¸ A¹ Aù Aú B' B(B]
B^ B| B} BŒ B B—
B˜ Bš B¸ B¹ B¾ B¿ BÀ BÁ BÚ Bà Bâ Bã Bî Bò Bó Bõ
Bö C C C C C C C, C2 C4 C5 CN CR CS CU CV
Cx C{ C| C} C~ C C′ C˜ Cš C› C³ C¸ C¹ Cº C»
CÌ CÑ CÒ CÓ CÔ D D D D D
D! D& D' D(D) DC DG DH DJ DK Dw D{ D| D~ D D
¤ D© Dª D« D¬ D³ D¸ D¹ Dº D»
DÙ Dß Dá Dâ Dò Dø Dù Dú Dû E
E
E E E E EA EF EG EH EI E_ Ed Ee Ef Eg E E…
E† Eˆ E‰
Eµ E¹ Eº E¼ E½ Eâ Eç Eè Eé Eê Eû F F F F

F F F F& F+ F, F- F. F9 F= F>
F@ FA FS FY F[F\ Fk Fl Fm Fn Fƒ Fˆ F‰
FŠ F‹ F™ Fš F› Fœ F Fž FŸ F F¡ F£ F© F« F¬ F»
F¼ F½ F¾ F¿ FÀ FÁ FÂ FÅ FÊ FË FÌ FÍ FØ FÙ FÚ FÛ
FÜ FÝ FÞ Fß Fà Fá Fä Fê Fì Fí Fù Fú Fû Fü Fý Fþ
Fÿ G G G G G G
G
G

G G G G G! G" G# G& G, G. G/ G; G< G= G> G?
G@ GA GB GC GD GF GI GJ GK GL GM GZ G[G\ G]
G^ G_ G` Ga Gb Ge Gi Gj Gl Gm G„ G… G† G‡ G‹ GŒ
G GÀ GÁ Gï Gð H# H$ H% H& H' H-
H. H3 H4 H7 H8 H; H< HF HG HI H` Hb Hg Hh Hi Hj
HŸ H H¡ H¢ H£ H¤ H
¥ H× HØ HÙ HÚ HÛ HÜ HÝ I I I I I I I IP IQ
IR IS IT IU IV IŠ I‹ IŒ I IŽ I I IÄ IÅ IÆ IÇ
IÈ IÉ IÊ Iù Iú Iû Iü Iý Iþ Iÿ J J
J
J

J
J J J J J J J J" J' J(J)
JI JJ JT JU JV JW JX JY JZ J“ J” J• J– J—
J˜ J™ JÄ JÅ JÆ JÇ JÈ JÉ JÊ JÜ JÝ JÞ Jß Jà Já Jâ
Jã Jí Jî Jï Jð Jö J÷ Jø Jü Jý Jþ Jÿ K K K K
K8 K9 K= K> K?
K@ KA KB KC Kv Kw Kx Ky Kz K{ K| K°
K± K² K³ K´ Kµ K¶ KÍ KÎ KÏ KÐ KÑ KÒ KÓ KÔ KÞ Kß
Kà Kò Kó Kô K÷ Kø L L L L
L L/ L0 L1 L2 L3 L4 L5 L6 L@ LA LB LG LH LI LN
LQ LR LS LT LŒ L LŽ L L L‘ L’ LÇ LÈ LÉ LÊ LË
LÌ LÍ Lò Ló Lô Lõ Lö L÷ Lø Lù M M M M M
M
M

M□ M□ M□ M□ M□ M M M$ M' M(M)
M* ML MM MN MO MP MQ MR MS M]
M^ M_ Mb Mc Md Me Mj Mq Mr M– M— M« M¬ M-
M® M¯ M°
M± Mç Mè Mé Mê Më Mì Mí N N□ N N□ N□ N□ N□ N
N□ N□ N□ N□ N□ N N N N* N+ N5 N: N; N< Nt Nu
Nv Nw Nx Ny Nz N»
N¼ N½ N¾ N¿ NÀ NÁ NÂ NÌ NÍ NÒ NÓ NÔ NÕ NÖ NØ NÙ
NÚ NÛ NÜ Nà Næ Nç Nè O
O□ O□ O□ OK OO OP OQ OR O{ O| O} O~ O O
€ O O, OŒ O O’ O“ O” O• O– Oš O¡ O¢
O× OØ OÙ OÚ OÛ OÜ OÝ P□ P□ P□ P□ P□ P□ P□ PK PL
PM PN PO PP PQ Pq Pr Ps Pt Pu Pv Pw Px P, P*f* P“
P” P• P– P— P™ Pš P
¥ P© Pª P« P¬ PÞ Pß Pà Pá Pâ Pã Pä Q□ Q□ Q□ Q□ Q□
Q□ Q□ Q* Q+ Q, Q-
Q. Q/ Q0 Q1 Q; Q< Q= QA QB QC QF QG QH QI QP QV
QW QX Q‰ QŠ Q‹ QŒ Q QŽ Q Q Qš Q› Q£ Q¤ Q
¥ Q¬ Q² Q³ Q´ Qã Qä Qå Qæ Qç Qè Qé Qê Qô Qõ Qþ Qÿ
R R□ R
R□ R□ R□ RE RF RG RH RI RJ RK R~ R R
€ R R, R*f* R„ R· R¸ R¹ Rº R»
R¼ R½ RÉ RÊ RË RÌ RÍ RÎ RÏ RÐ RÚ RÛ Rá Râ Rã Rä
Rå Rê Rñ Rò S' S(S) S* S+ S, S-
SJ SK S_ S` Sa Sb Sc Sd Se S• S– S— S˜ S™ Sš S›
Sª S« S¬ S S® S¯ S° S± S»
S¼ SÉ SÊ SË SÛ Sß Sà Sá Sâ T□ T□ T□ T□ T□ T□ T
TN TO TP TQ TR TS TT T‡ Tˆ T‰ TŠ T‹ TŒ T T-
T® T¯ T°
T± T² T³ T´ T¾ T¿ TÀ TÅ TÆ TÇ TÈ TÉ TÊ TÎ TÓ TÔ
TÕ U
U

U
U U U U UC UD UE UF UG UH UI U} U~ U U
€ U U, Uƒ U² U³ U´ Uµ U¶ U·
U¸ UÏ UÐ UÑ UÒ UÓ UÔ UÕ UÖ Uà Uá Uí Uî Uï Uû Uü
Uý Uþ Uÿ V V V V V V V V V V V V# V%
V& V' V(V) V\ V] V^ V_ V` Va Vb V– V—
V˜ V™ Vš V› Vœ VÓ VÔ VÕ VÖ V× VØ VÙ W

W
W W W W W W W W W W W W W
WQ WR WS WT WU WV WW Wq Wr Ws Wt Wu Wv Ww Wx Wy
Wz W{ Wª W« W¬ W W® W¯ W°
W± W² W³ W´ WÁ WÂ WÃ WÄ WÅ WÆ WÇ WÈ WÊ WË WÌ WÍ
WÎ WÏ WÐ WÑ WÒ WÓ WÔ Wâ Wã Wä Wå Wæ Wç Wè Wë Wì
Wí Wî Wï Wð Wñ X' X(X) X* X+ X, X-
X` Xa Xb Xc Xd Xe Xf Xš X›
Xœ X Xž XŸ X XÔ XÕ XÖ X× XØ XÙ XÚ Y Y Y Y
Y Y Y YE YF YG YH YI YJ YK Yp Yq Yr Ys Yt Yu
Yv Yw Y Y, Y• Y– Y— Y˜ Y™ Yš Y›
Yœ Y Yž YŸ Y Y¡ Y¢ Y± Y² Y³ Y´ Yµ Y¶ Y·
Y¸ Y¹ Yº Y»
Y¼ Y½ Y¾ YÎ YÓ YÔ YÕ Z Z Z Z Z Z Z ZI ZJ
ZK ZL ZM ZN ZO ZW ZX Zƒ Z„ Z… Z† Z‡ Zˆ Z‰
Z® Z¯ Z°
Z± Z² Z³ Z´ Zµ ZÙ ZÚ ZÛ ZÜ ZÝ ZÞ Zß Zà Zá Zâ Zã
Zä Zå Zö Z÷ Zø Zù Zú Zû Zü Zý Zþ Zÿ [[[[
[[[[[[[[[[[, [1 [2 [3 [
h [i [j [k [l [m [n [ƒ [„ […] [† [‡ [ˆ [‰ [Š [”
[• [š [›
[œ [[ž [Ÿ [[¤ [¨ [© [ª [« [á [â [ã [ä [å [æ
[ç \

\
\ \ \ \ \ \ \ \ \# \$ \% \& \' \, \2
\3 \4 \j \k \l \m \n \o \p \¨ \© \ª \« \¬ \ \®
\¯ \¹ \º \Â \Ã \Ä \Å \Í \Î \Ï \Ð \Ñ]]]
]7]8]9]l]m]n]o]p]q]x]y]z]]€]…]
†]Š]‹]Ž]]’]“]]ž]]µ]ø]ù]ý]þ ^4 ^
5 ^6 ^7 ^8 ^p ^q ^r ^s ^t ^Ÿ ^ ^¡ ^¢ ^£ ^¤ ^
¥ ^¦ ^§ ^¶ ^· ^»
^¼ ^ð ^ñ ^ò ^ó ^ô ^õ ^ö ^÷ ^ø _ _ _ _ _U _V
_W _X _Y _ _, _† _‡ _¸ _¹ _º _»
_¼ _Þ _ß _à _á _â _ú _û _ü _ý _þ _ÿ ` ` ` `
` ` ` ` ` `1 `
2 `k `l `¦ `§ `â `ã a a aM aN a† a‡ aÄ aÅ b b
b5 bi bj b¤ b
¥ bß bà bá bâ c c c c cT cU cV cW c c‘ c’ c“
cª c« c¬ c-
c® c¾ c¿ cÀ cÁ cÂ cß cà cá câ cã cå cæ cç cè cé
cê cë cì cí cî cï cð cñ d d
d
d

d
d d d d d d$ d% d& d' d* d+ d, d-
d. d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 d: d; dQ dR dU
dV dW dX dY dv dw dx dy dz d
€ d d, dƒ d„ d… d dž dŸ d d¡ d¢ d
£ d© dª d« d¬ d d® d¯ d°
d¿ dÀ dÁ dÂ dÃ dÄ dÅ dÆ dÇ dÉ dÊ dË dÌ dÍ dÎ dÏ
dÐ dÖ d× dØ dÙ dÚ dÛ dÜ dÝ dÞ dà dá dâ dã dä då
dæ dë dì dí dî dï d÷ dø e4 e5 e6 e7 eO eP eQ eR e
S eT eU eV ec ed e‘ e’ e“ e” e• e– e—
eÃ eÄ eÅ eÆ eÇ eÈ eÉ eü eý eþ eÿ f f f f f
f f f f fX fY fZ f[f\ f’ f“ f” f• f–
fÇ fÈ fÉ fÊ fË fÌ fÍ fÎ fà fá fæ fç fè fé fê g
g g! g" g# g[g\ g] g^ g_ g~ g g´ gµ g¶ g·
gì gí gî gï gð gñ gò gó h

h
h h h h h h h h. h/ h0 h1 h2 h3 h4 h5 hd
he hf hg hh hi hj h›
hœ h hž hŸ h h¡ hÐ hÑ hÒ hÓ hÔ i i i
i
i

 i= i> i? i@ iA iz i{ i| i} i~ i® i¯ i°
 i± i² ié iê ië iì ií j" j# j$ j%
 j& j1 j2 j3 j4 j5 j6 j7 j8 j9 jK jL jV jW jY jp
jq j® j¯ jî jï k, k- kb kc kš k›
 kœ k   k¹ kº kÍ kÎ kÏ kÐ kö k÷ kø kù kú kû kü ký
l
 l

l< l= l> l? l@ lA lB lC la lb lc ld l™ lš l›
lœ lÆ lÇ lÈ lÉ lÊ lË lÌ lÍ lÎ lÏ læ lç lè lé m
m m m!
m. m/ m0 m1 m2 m3 m4 m5 mD mE mF mG mH mI mJ mK
mL mM mN mO mP m\ m]
m^ m_ m` ma mb mc md me mf mg mn mo mp mq mr ms
mt mu m¦ m§ m¨ m© mª m« m¬ m mß mà mê më mî mï n
n) n* nC nb nc n™ nš n
¥ n¦ n© nª nÈ nç nè o& oC ob oc ož oß oñ oò p/ p4
p5 p^ p_ pd pe € € @€ A€ € S
€ m€ n€ € €€ € z] € € ¿€€ € z^ Ý
€ ç€ <€ f€ –€ ®€ Ç€ Ð€
€)€€ € za € -€€ € zb € ƒ€€ € z € ³€€ € z,
€ 2€€ € z¡ € zÀ € ï€€ € zß € €€ € zà € %€
€ zá :€€ € zâ zî €€ € zï € C€€ € zð
€ …€€ € zñ € ¡€€ € zò € Ç€€ € zó €

€ € zô € -€ € zõ € 7€ € zö € z÷ € =€
€ zø € M€ € zù € ,€ € zú € zû € ¯€ € {
€ Ò€ € { € ÷€ € { € € € {
€ ` € {
€ ¦ € {

€ ½€ € É€ € {
€ Í€ € { € é€ € { € +€ € { € o€
€ { € ^€ € { € °€ € { € ò€ € { €
7€ € { €
}€ € {
Ä€ € { €
'€ € { €
d€ € { €
h€ € { €
x€ € { €
©€ € { €
ê€
ú€

,€

.€

H€ € { €

Š€ € { € { €

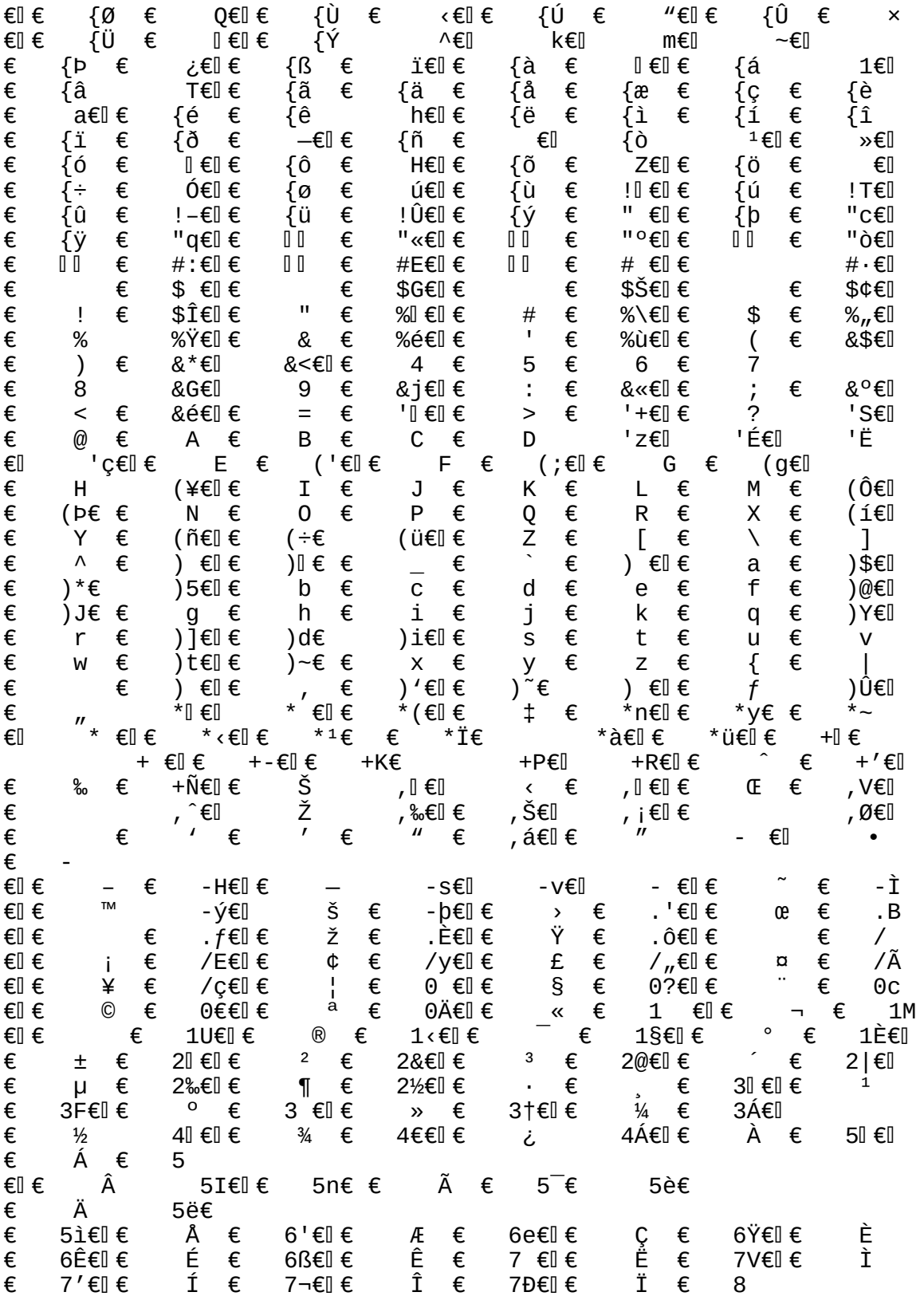
¬€ € {R €

Í€ € {S €
 € € {V €

€□€ {W €
P€□€ {X €
€□€ {Y €
Ÿ€□€ {Z €
Ò€□€ {[€
æ€□€ {` € □□€□€ {c € □5€□€ {d € □O€□€ {e
€ □’€□€ {f € □Ð€□€ {g € □
€□€ {h € □□€□€ {i € □Y€□€ {j € □•€□€ {k € □¬
€□€ {l € □Ï€□€ {m € □□€□€ {n € □!€□€ {o € □S
€□€ {p € □•€□€ {q € □Ñ€□€ {r € □Ø€□€ {s € □
€□€ {t € □'€□€ {x € □.€□€ {y € □:€□€ {z € □=
€□€ {| € □[€□€ {} € □h€□€ {~ € □ž€□€ { □Ý
€□€ {€ € □ €□€ { € □I€□€ {, € □…€□€ {ƒ
€ □‹€□€ {„ € □Æ€□€ {… € □Ô€□€ {‰ € □ø€□€ {Š
€ □□€□€ {‹ □I€□€ {Œ € □±€□€ { € □Ø€□€ {Ž
€ □□€□€ { € □<€□€ { € □Z€□€ {‘ € □|€□€ {’
€ □œ€□€ {“ € □Ò€□€ {” € □â€□€ {• € □'€□€ {–
€ □2€□€ {— € □j€□€ {˜ € □°€□€ {™ € □ï€□€ {š
€ □2€□€ {› € □?€□€ {œ € □t€□€ { € □¢€□€ {ž
€ □¹€□€ {Ÿ € □ÿ€□€ { € □ €□€ {¡ € □@€□€ {¢
€ □†€□€ {£ € □Ä€□€ {¤ € □É€□€ {¥ € □

€□ € {¦ € □□€□ € {§ € □□€□ € {¨ € □S€□ € {© € □^
€□ € {ª □,€□ € {¯ € □´€□ € {° € □Ã€□ € {± € □Ó
€□ € □ê€□ € {³ € □ù€□ € {· € □□€□ € {¸ € {Å € □H
€□ € {Æ € □R€□ € {Ç € □‹€□ € {È € □Ï€□ € {É € □

€□€ {Ê € □□€□€ {Ë € □V€□€ {Ì € □Ž€□€ {Í
€ □—€□€ {Î € □Ù€□€ {Ï □□€□€ {Ð € □□€□€ {Ñ
€ □Ä€□€ {Ò € □€□€ {Ó € F€□€ {Ô € ^€□€ {Õ
€ ²€□€ {Ö € É€□€ {× €



€ € Ð 8S€ € ï € 8“€ € ð € 8Ë€ € ñ € 9
€ € ò € 9B€ € ó € 9,€ € ô € 9›€ € õ € 9Á
€ € ö € 9ß€ € ÷ 9ý€ € ø € :y€ € ù € :¾
€ € ú € :ÿ€ € û € ;&€ € ü € ;B€ € ý € ;R
€ € þ ;^€ € ÿ € ‘ € ‘ € ‘ € ‘ € ;Ÿ
€ ;¡€ € ‘ € ‘ € ‘ € ‘ € ‘ €
€ ‘
€ ‘
€ ;Ì€ € ‘

;ò€ € ‘
€ ‘ € ‘ € ‘ € ‘ € ‘ € ‘ € ‘ € ;ò€
<

```
€      <C€      <N€ €    <}€ €    ‘  €    <•€ €    ‘  €    <Ã€
€    ‘  €    <Ñ€ €    ‘  €    =  € €    ‘  €    =S€ €    ‘  €    = €
€    ‘  €    =˜€ €    ‘   €    =Ü€ €    ‘   €    =å€ €    ‘   €    >  €
€    ‘   €    >!€ €    ‘   €    >L€ €    ‘!  €    >e€ €    ‘"  €    >©€
€    ‘#  €    >Ý€ €    ‘$  €    >í€ €    ‘%  €    ?2€ €    ‘&  €    ?;€
€    ‘'  €    ?v€ €    ‘(  €    ? €€    ‘)  €    ?º€ €    ‘*  €    @ €
€    ‘+       @5€      @7€ €    ‘,  €    @~€ €    ‘-  €    @Ã€ €    ‘.
€    A
€ €    ‘/  €    A(€ €    ‘0  €    AL€ €    ‘1  €    A €€    ‘2  €    AÔ
€ €    ‘3  €    B€ €    ‘4       Ba€ €    ‘5  €    BÃ€      BÍ€
```

```
    BÎ€      Bß€      Bñ€ €   ‘6  €   C%€ €   ‘7  €   CY€
€   ‘8       C €€    ‘9       Cý€ €   ‘:       D
€€   ‘;  €    ‘<       D'€ €    ‘=  €    ‘>  €    ‘?  €    ‘@
€   ‘A  €   ‘B  €   ‘C  €   ‘D  €   ‘E       DQ€ €   ‘F  €   ‘G
€   ‘H  €   ‘I  €   ‘J  €   ‘K  €   ‘L  €   ‘M  €   ‘N       Dp€
€   ‘O  €   ‘P  €   ‘Q  €   ‘R  €   ‘S  €   ‘T  €   ‘U  €   ‘V
€   ‘W       D‘€ €   ‘X  €   ‘Y  €   ‘Z  €   ‘[  €   ‘\  €   ‘]
€   ‘^  €   ‘_  €   ‘`  €   ‘a  €   ‘b       D²€ €   ‘c  €   ‘d
€   ‘e  €   ‘f  €   ‘g  €   ‘h  €   ‘i  €   ‘j  €   ‘k  €   ‘l
€   ‘m       DÔ€ €   ‘n  €   ‘o  €   ‘p  €   ‘q  €   ‘r  €   ‘s
€   ‘t  €   ‘u  €   ‘v  €   ‘w  €   ‘x       E7€ €   EK€
€   ‘y       Ec€      Ex€ €   ‘z  €   E³€ €   ‘{  €   EÕ€
€   ‘|       F$€ €   ‘}  €   F\€ €   ‘~       F|€ €   ‘
€   F¡€      F«€
```

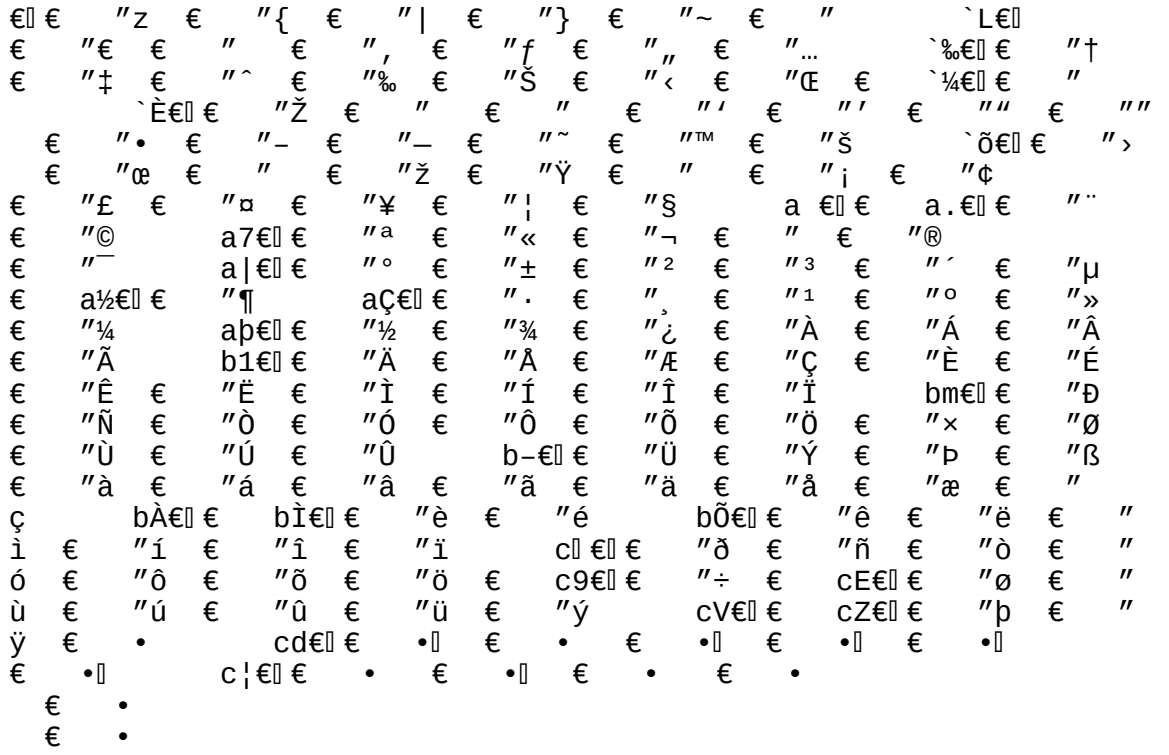
	F¬€	F¸ €	FÖ€ €	F×€ €	‘€ €	FÞ€ €	‘	Fá€
€	Fú€ €	G € €	‘,	G € €	G € €	‘ƒ €	G €
€	‘„	G € €	G6€ €	‘… €	‘† €	G=€ €	‘‡	G@€
€	GR€ €	GX€ €	‘ˆ	G\€ €	Gu€ €	‘‰ €	G{€
€	‘Š	G € €	G¡€ €	‘‹ €	‘Œ €	G¨€ €	‘	G«€
€	G¾€ €	GÄ€ €	‘Ž	GÈ€ €	Gà€ €	‘ €	Gç€
€	‘	Gê€ €	Gû€ €	‘’ €	H € €	‘’	H € €	H3€
€	‘“ €	H9€ €	‘”	H=€ €	HT€ €	‘• €	H[€ €	‘–
	H^€ €	Hx€ €	‘— €	H~€ €	‘˜	H,€ €	H®€
€	‘™ €	H´€ €	‘š	H¸ €	HÝ€ €	‘› €	Hä€
€	‘œ	Hç€ €	Hî€ €	‘ €	Hõ€ €	‘ž	Hø€ €	I €
€	I € €	‘Ÿ	I € €	I0€ €	I6€ €	I7€ €	‘ €	I:
€	II€ €	IJ€ €	‘¡ €	IQ€ €	‘¢	IT€ €	I,€
€	‘£ €	I‰€ €	‘¤	IŒ€ €	I¢€ €	‘¥ €	I©€
€	‘¦	I¬€ €	IÆ€ €	‘§ €	IÌ€ €	‘¨	IÐ€ €	Iü€
€	‘© €	J € €	‘ª	J € €	J+€ €	‘« €	J2€
€	‘¬	J5€ €	JF€ €	JL€ €	‘	JP€ €	JX€ €	J^€
€	‘®	Jb€ €	Js€ €	‘¯ €	Jz€ €	‘°	J}€ €	Jˆ€
€	‘± €	JŽ€ €	‘²	J’€ €	J¤€ €	Jª€ €	‘³ €	J®€
€	‘´ €	‘µ €	‘¶	JÃ€ €	JØ€ €	‘· €	Jß€ €	‘¸
€	Jâ€ €	‘¹ €	‘º €	‘» €	‘¼ €	‘½ €	‘¾ €	‘¿
€	‘À	K € €	K € €	K € €	‘Á €	K

```
€□€   ‘Â  €   ‘Ã  €   ‘Ä  €   ‘Å  €   ‘Æ  €   ‘Ç
€   ‘È      K)€□€   K,€□€   ‘É  €   K3€□€   ‘Ê  €   K6€□€   ‘Ë
€   ‘Ì  €   ‘Í  €   ‘Î  €   ‘Ï  €   ‘Ð  €   ‘Ñ  €   ‘Ò
€   ‘Ó      KS€□€   KV€□€   K\€□€   ‘Ô  €   K`€□€   ‘Õ  €   ‘Ö
€   ‘×  €   ‘Ø  €   ‘Ù  €   ‘Ú  €   ‘Û  €   ‘Ü  €   ‘Ý      K~€□
€   K€€□€   ‘Þ  €   K‡€□€   ‘ß  €   KŠ€□€   ‘à  €   ‘á  €   ‘â
€   ‘ã  €   ‘ä  €   ‘å      K§€□€   Kª€□€   K°€□€   ‘æ  €   K´€□
€   ‘ç  €   ‘è  €   ‘é  €   ‘ê  €   ‘ë  €   ‘ì  €   ‘í  €   ‘î
€   ‘ï      KÒ€□€   KÔ€□€   ‘ð  €   ‘ñ  €   KÛ€□€   ‘ò  €   KÞ€□
€   ‘ó  €   ‘ô  €   ‘õ  €   ‘ö  €   ‘÷  €   ‘ø  €   ‘ù
€   ‘ú      Kû€□€   Kþ€□€   ‘û  €   L□€□€   ‘ü  €   L□€□€   ‘ý
€   ‘þ  €   ‘ÿ      L%€□    L+€□€   ’   €   L/€□€   ’□  €   Lf€□
€   ’       L˜€□€   ’□  €   ’□  €   ’□  €   ’□  €   LÓ€□€   ’
€   LÛ€□€   ’□  €   Lâ€□€   ’          Lç€□€   ’
  €   M□€□    M□€
```

```
      T€▯€   TÏ€▯€   “
  €   “▯      TÙ€▯€   U▯€▯€   “▯   €   “▯   €   “▯      U €▯
€   “▯   €   “▯   €   “▯   €   “▯   €   “▯   €   “▯   €   “▯   €   US€▯
€   “▯   €   U_€▯€   “▯   €   “▯   €   “   €   “       Ul€▯€   Up€▯
€   “       Uy€▯€   “   €   “   €   “!   €   “"   €   “#
€   “$      U°€▯€   “%   €   “&   €   “'   €   “(   €   “)
€   “*      Uû€▯€   “+   €   “,   €   “-   €   “.   €   “/
€   “0      V:€▯€   “1   €   “2   €   “3   €   “4   €   “5   €   “6
€   “7   €   Vh€▯€   “8   €   Vt€▯€   “9   €   “:   €   “;   €   “<
€   VŒ€▯€   “=      V €▯€   V›€▯€   “>   €   “?   €   “@      V¥€▯
€   “A   €   “B   €   “C   €   “D   €   “E   €   “F      Vã€▯€   “G
€   “H   €   “I   €   “J   €   “K   €   “L      W €▯€   “M   €   “N
€   “O   €   “P   €   “Q   €   “R   €   “S   €   W=€▯€   “T   €   “U
€   WK€▯€   “V   €   “W   €   WS€▯€   “X   €   “Y   €   “Z      W\€▯
€   Wc€▯€   “[   €   “\      Wm€▯€   “]   €   “^   €   “_   €   “`
€   “a   €   “b   €   “c   €   W¬€▯€   “d   €   W¸ €▯€   “e
€   “f      WÄ€▯€   WË€▯€   “g   €   “h      WÕ€▯€   “i   €   “j
€   “k   €   “l   €   “m   €   “n   €   “o   €   X▯€▯€   “p   €   X €▯
€   “q   €   “r   €   “s      X-€▯€   X5€▯€   “t   €   “u      X>€▯
€   “v   €   “w   €   “x   €   “y   €   “z   €   “{      X €▯€   “|
€   “}   €   “~   €   “   €   “€   €   “       X¾€▯€   “,   €   “ƒ
€   “„   €   “…   €   “†   €   “‡      Xý€▯€   “ˆ   €   “‰   €   “Š
€   “‹   €   “Œ   €   “   €   “Ž   €   Y▯€▯€   “   €   Y#€▯€   “
€   “‘   €   “’   €   ““      Y1€▯€   Y6€▯€   “”      Y?€▯€   “•
€   “–   €   “—   €   “˜   €   “™   €   “š   €   Y€€▯€   “›      YŸ€▯
€   “œ   €   “   €   “ž   €   “Ÿ   €   “   €   “¡      Y¿€▯€   “¢
€   “£   €   “¤   €   “¥   €   “¦   €   “§      Yû€▯€   “¨   €   “©
€   “ª   €   “«   €   “¬   €   “   €   “®   €   Z▯€▯€   “¯   €   Z$€▯
€   “°   €   “±      Z5€▯€   ZE€▯€   “²   €   “³   €   “´      ZO€▯
€   “µ   €   “¶   €   “.   €   “   €   “¹   €   “º      Z €▯€   “»
€   “¼   €   “½   €   “¾   €   “¿   €   “À      ZÍ€▯€   “Á   €   “Â
€   “Ã   €   “Ä   €   “Å   €   “Æ      [
```

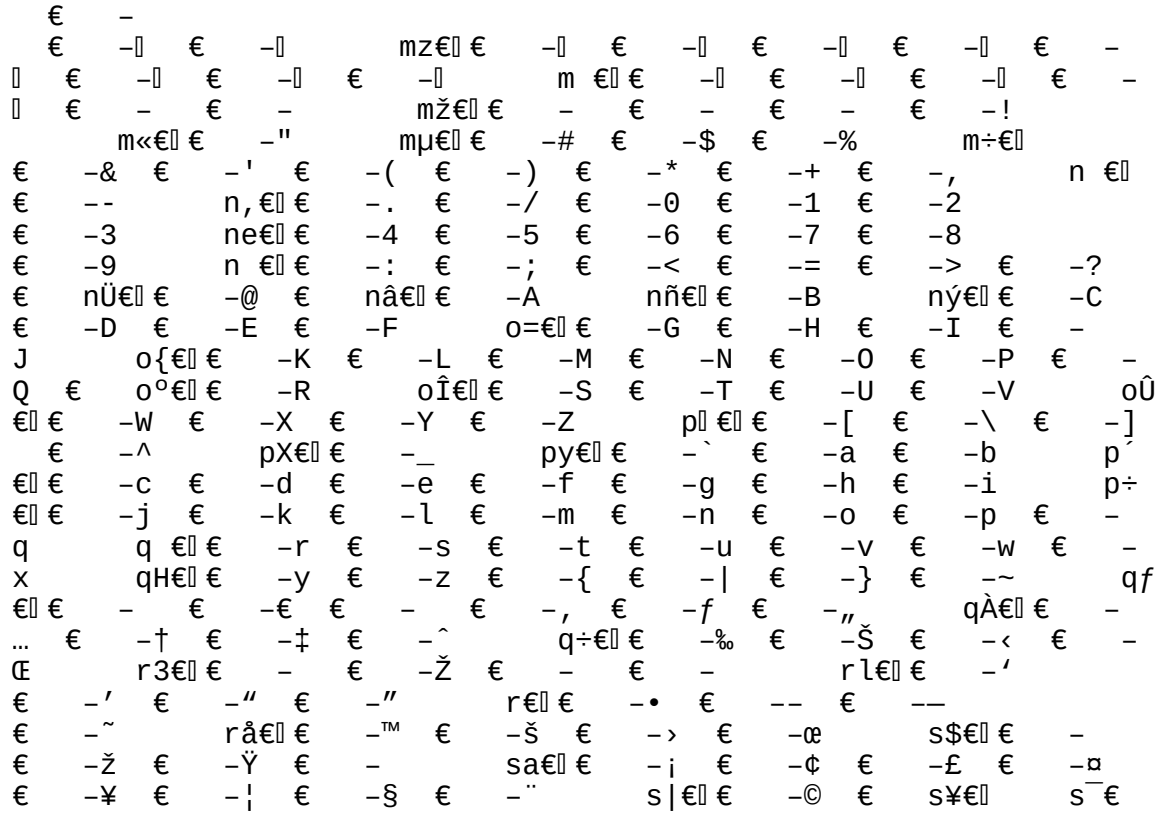
```
€□€   “Ç  €   “È  €   “É  €   “Ê  €   “Ë  €   “Ì  €   “Í  €   [:
€□€   “Î  €   “Ï  €   [H€□€   “Ð  €   “Ñ  €   “Ò  €   “Ó
€   “Ô      [W€□€   [[€□€   “Õ  €   “Ö      [d€□€   “×  €   “Ø
€   “Ù  €   “Ú  €   “Û  €   “Ü      [¦€□€   “Ý  €   “Þ  €   “ß
€   “à  €   “á  €   “â      [ä€□€   “ã  €   “ä  €   “å  €   “æ
€   “ç  €   “è      \$€□€   “é  €   “ê  €   “ë  €   “ì  €   “í
€   “î      \_€□€   “ï  €   “ð  €   “ñ  €   “ò  €   “ó  €   “ô
€   “õ  €   \„€□€   “ö  €   \ €□€   “÷  €   “ø      \ €□€   “ù
€   “ú  €   “û  €   “ü  €   “ý  €   “þ  €   “ÿ  €   ”   €   ”□
€   ”   €   ”□  €   ”□  €   ”□  €   \Æ€□€   ”□      \Ö€□€   \â€□
€   ”   €   ”□  €   ”   €   ”
      \ì€□€   ”
  €   ”
```

```
  €   ”
  €   ”▯  €   ”▯  €   ”▯       ]+€▯€   ”▯  €   ”▯  €   ”▯  €   ”
▯  €   ”▯  €   ”▯       ]k€▯€   ”▯  €   ”▯  €   ”▯  €   ”▯  €   ”
▯  €   ”       ]®€▯€   ”   €   ”   €   ”   €   ”   €   ”!
€   ””      ]í€▯€   ”#  €   ”$  €   ”%  €   ”&  €   ”'  €   ”(
€   ”)  €   ”*      ^▯€▯€   ”+  €   ”,  €   ”-  €   ”.  €   ”/
€   ”0      ^@€▯€   ”1  €   ”2  €   ”3  €   ”4  €   ”5  €   ”6
€   ”7  €   ”8  €   ”9  €   ”:      ^n€▯€   ”;  €   ”<  €   ”=
€   ”>  €   ”?  €   ”@  €   ”A  €   ”B  €   ”C  €   ”D  €   ^±€▯
€   ”E  €   ”F  €   ”G  €   ”H  €   ”I  €   ”J  €   ”K      ^Ì€▯
€   ”L  €   ”M  €   ”N  €   ”O  €   ”P  €   ”Q  €   ”R  €   ”S
€   ”T  €   ”U  €   ^â€▯€   ”V  €   ”W  €   ”X  €   ”Y  €   ”Z
€   ”[      ^ü€▯€   ”\  €   ”]  €   ”^  €   ”_  €   ”`  €   ”
a       _
€▯€   ”b  €   ”c  €   ”d  €   ”e  €   ”f  €   ”g      _M€▯€   ”
h  €   ”i  €   ”j  €   ”k  €   ”l  €   ”m      _Œ€▯€   ”n  €   ”
o  €   ”p  €   ”q  €   ”r  €   ”s      _Ì€▯€   ”t  €   ”u  €   ”
v  €   ”w  €   ”x  €   ”y      `
```



```
  €   •
  €   cÙ€▯€   •▯  €   cå€▯€   •▯  €   •▯  €   •▯  €   •▯      cò
€▯€   c÷€▯€   •▯  €   •▯      d▯€▯€   •▯  €   •▯  €   •▯
€   •▯  €   •▯  €   •▯      dC€▯€   •▯  €   •   €   •   €   •
€   •   €   •   €   •!  €   d‰€▯€   •"  €   d•€▯€   •#  €   •$
€   •%      d£€▯€   •&  €   •'  €   •(  €   •)      d³€▯€   •*
€   •+      dë€▯€   •,  €   •-      e €▯€   •.  €   •/  €   •0
€   •1  €   •2  €   e\€▯€   •3  €   •4  €   eg€▯€   •5  €   en€▯
€   •6  €   eu€▯€   •7  €   e{€▯€   •8      e€€▯€   •9
€   e›€▯    e¥€
    e¦€▯    e·€▯    eÌ€▯€   •:  €   f▯€▯€   •;      f▯€▯€   •<
€   •=  €   •>  €   •?      fU€▯€   •@  €   •A  €   •B
€   •C      f•€▯€   •D  €   •E  €   •F  €   •G  €   •H  €   •I
€   •J  €   •K      fÐ€▯€   •L  €   fá€▯€   •M      fç€▯€   •N
€   •O  €   •P  €   •Q  €   •R  €   •S  €   •T  €   •U      g+€▯
€   •V  €   gL€▯€   •W      gR€▯€   •X  €   •Y  €   •Z
€   •[      g’€▯€   •\  €   g¼€▯€   •]      gÂ€▯€   •^  €   •_
€   •`  €   •a      gû€▯€   •b  €   •c  €   •d  €   •e      h%€▯
€   •f  €   •g  €   •h  €   •i  €   •j  €   •k  €   •l  €   •m
€   •n      hO€▯€   •o  €   hr€▯    h|€
```

h}€ hŒ€ h¡€ € h£€ € •p € hÝ€ € •q € i€ €
€ •r € iT€ € •s € iˆ€ € •t € i¿€ € •u € iø€
€ •v € j6€ € •w js€ € j¨€ € •x jÝ€
€ •y k€ € € •z € •{ € •| kY€ € •} € •~
€ • k˜€ € € •€ € • € •¸ kÔ€ € •ƒ € •„
€ •… l€ € € •† € •‡ € •ˆ € •‰ l2€ € € •Š
€ •‹ € •Œ € • lJ€ € •Ž € • € • € •‘
€ lo€ € •’ € •“ € •” € •• € •– € •— € •˜
€ •™ € •š € •› € •œ € • € •ž l‰€ € •´
€ •µ € •¶ € •· € l”€ € •¸ € l˜€ € •¹ € •º
€ •¾ € •¿ € •À l®€ € •Á € •Â € •Ã € •Ä
€ l¹€ € •Å € •Æ € •Ç € •È € •É € •Ê € •Ë
€ •Ì € •Í € •Î € •Ï € •Ð € •æ lÓ€ € •ç
€ •è € •é € •ê lÞ€ € •ë € •ì € •í
€ •î m€ € •ï € •ð € •ñ € •ò € •ó m€
€ •ô € •õ € •ö € •÷ € •ø € •ù m7€ € •ú
€ •û € •ü € •ý € •þ € •ÿ € – mK€ € –
€ – € – € – € – € – € – € – mj€
€ – € –
€ –
€ –



s°€▯ sÂ€▯ sÙ€▯€ sÜ€▯€ –ª € t¯▯€▯€ –« € tZ€▯
€ –¬ t˜€▯€ – tÏ€▯€ –® € –¯ € –° € u

```
€ € –± u*€ € –² € –³ € –´ uC€ € –µ
€ –¶ € –· € –¸ € –¹ € –º € –» uw€ € –
¼ u‡€ € –½ € –¾ € –¿ € –À € –Á € –Â € –
Ã uÅ€ € –Ä € –Å € –Æ ué€ € –Ç € –È € –
É v$€ € –Ê € –Ë € –Ì € –Í € –Î € –Ï € –
Ð € –Ñ € –Ò € v`€ vw€ vx€ € –Ó v{€ € –
Ô € –Õ € –Ö v¶€ € –× € –Ø € –Ù € –Ú € –
Û € –Ü € –Ý vÑ€ € –Þ € –ß € –à € –á € –
â € –ã € –ä € –å € –æ € –ç € –è € –é vø
€ € –ê € –ë € –ì € –í € –î € –ï € –ð € –
ñ € –ò € –ó € –ô w€ € –õ € –ö € –÷ € –
ø € –ù € –ú € –û w/€ € –ü € –ý € –þ € –
ÿ € — € —€ — wn€ € —€ € w£€ w€ w®
€ € —€ € w³€ € —€ € —$ € wó€ € —% € —D € x1
€ € —E € xm€ xx€ xy€ € —F € x~€ € —G € —
f € xÁ€ € y€ € —g € —† € yF€ € y†€ yÌ€ € —
‡ € yà€ € z"€ € —ˆ z)€ —‰ zT zY
€
```