

Tasmanian Secondary Assessment Board

Tasmanian Certificate of Education

External Assessment

1999

IF862 COMPUTER SCIENCE

Time: Three Hours

On the basis of your performance in this examination, the examiners will provide a rating of A, B, C or D on each of the following criteria taken from the syllabus statement:

- Criterion 5** Demonstrate a knowledge of the Pascal programming language.
- Criterion 6** Interpret descriptions of language syntax.
- Criterion 8** Manipulate expressions in a formal logic (*essential*).
- Criterion 10** Represent basic data types using fixed-length binary representations.
- Criterion 11** Demonstrate familiarity with a simple computing machine.

Pages: 13
Questions: 15
Attachment: Logic Laws

© Copyright for part(s) of this examination may be held by individuals and/or organisations other than the Tasmanian Secondary Assessment Board.

BLANK PAGE

CANDIDATE INSTRUCTIONS

This paper contains five sections which assess Criteria 5, 6, 8, 10 and 11.

Each section consists of three questions.

Candidates should attempt **all** questions.

To be considered for a 'C' rating on a criterion candidates must provide a satisfactory answer to at least the first question of the relevant section. To be considered for a 'B' rating on a criterion candidates must provide satisfactory answers to at least the first two questions of the relevant section. To be considered for an 'A' rating on a criterion candidates must provide satisfactory answers to all three questions of the relevant section.

Candidates should make sure that **all five sections are attempted** in order to achieve a rating on **each criterion**.

Allocate approximately **35 minutes to each section** for this purpose.

Candidates should show the methods used in deriving answers.

Candidates should **take care with the presentation** of their answers which should be complete and to the point. Diagrams should be used where appropriate. Candidates are reminded that writing, spelling and expression which make it difficult for the examiner to understand what candidates mean will be taken into account in the assessment.

Candidates may bring up to five books and one folder of notes into the examination room. These may be consulted freely.

Calculators are **not** permitted in the examination room.

Section A: Criterion 5

All questions in this section relate to the program **opposite**. The numbers in the left margin are for reference only and are not part of the program. The program is designed to play a guessing game with the following rules:

The game uses 12 marbles, three of each colour – red, green, blue and white. A number of marbles are selected and the object of the game is to determine the number of each colour marble that has been chosen. After trying to guess the number of each coloured marble the player is told how many guesses were too high. Using this information the player tries again.

Question 1

(a) Calculate the value of the following expressions:

- (i) $5 + 4 / 1 * 2$
- (ii) $\text{sqr}(\text{sqr}(3))$
- (iii) $13 \bmod 4$

(b) Give the type for the constant *Max* declared as follows:

```
const Max = 10
```

(c) Give the output of the following code segment:

```
Finish := false;
Count := 0;
while not Finish do
begin
    Count := Count + 1;
    if Count > 5 then Finish := True
end;
write('Count =',Count:3)
```

Question 2

- (a) In the program when the variable *Finish* is *true* the player has won. Which procedure is used to determine whether the player has won? Explain how this procedure uses a loop to identify when the player has won.
- (b) Produce a diagram of the structure of the variable *Store*. Use this diagram to explain what is held in the variable *Store*.
- (c) Why is it necessary to use the variable *Count* in procedure *Print* rather than using the constant *Max*?

Question 3

- (a) If the *var* in front of the formal parameter *Finish* in the procedure *Check* was missing, what would always be last line of text output by the program? Explain your answer.
- (b) If the program was running and the screen display was as follows:

```
Please enter the number of each colour marble
Red? :0
Green? :3
Blue? :2
White? :3
```

What would be the next two lines output by the program?

```

1  Program Criterion5(Input, Output);
2  const   Max           = 10;
3  type    Colourtype   = (Red, Green, Blue, White);
4  Greatype = 0 .. 4;
5  Answertype = array[Red .. White] of 0 .. 3;
6  Storetype = array[1 .. Max] of record
7
8                      Answer : Answertype;
9                      Great : Greatype;
10
11                     end;
12 var      Correct : Answertype;
13          Store   : Storetype;
14          Count   : Integer;
15          Finish  : Boolean;
16 procedure Enter(var Guess : Answertype);
17 begin
18     writeln('Please enter the number of each colour marble');
19     write('Red? '); readln(Guess [Red]);
20     write('Green? '); readln(Guess [Green]);
21     write('Blue? '); readln(Guess [Blue]);
22     write('White? '); readln(Guess [White])
23 end;
24 procedure Initialise(var Correct : Answertype);
25 begin
26     Correct [Red] := 2; Correct [Green] := 0;
27     Correct [Blue] := 3; Correct [White] := 1
28 end;
29 procedure Check(Guess, Correct : Answertype; var Great : Greatype; var Finish : Boolean);
30 var
31     C : Colourtype;
32 begin
33     Great := 0; Finish := True;
34     for C := Red to White do
35     begin
36         if Guess[C] > Correct [C] then Great := Great + 1;
37         if Guess[C] <> Correct [C] then Finish := False
38     end
39 end;
40 procedure Print(Store : Storetype; Count : Integer);
41 var
42     I : integer;
43     C : Colourtype;
44 begin
45     writeln('Red':6, 'Green':6, 'Blue':6, 'White':6, 'Great':6);
46     for I := 1 to Count do
47     begin
48         for C := Red to White do
49             write(Store[I].Answer[C]:6);
50         writeln(Store[I].Great:6)
51     end
52 end;
53 begin
54     Initialise(Correct);
55     Finish := false;
56     Count := 0;
57     while not Finish and (Count < Max) do
58     begin
59         Count := Count + 1;
60         Enter(Store[Count].Answer);
61         Check(Store[Count].Answer, Correct, Store[Count].Great, Finish);
62         Print(Store, Count)
63     end;
64     if not Finish then writeln('Bad luck!') else writeln('Well done!')
65 end.

```

Section B: Criterion 6

Questions in this section refer to Grammar A and Grammar B given below.

Grammar A

$$\begin{aligned} S &::= P \nabla S \mid P \\ P &::= D \mid DD \mid DDD \\ D &::= \times \mid O \end{aligned}$$

Grammar B

$$\begin{aligned} S &::= D \nabla R \\ R &::= D \nabla D \mid D \\ D &::= DD \mid DDD \\ D &::= \times \mid O \end{aligned}$$

S is the Start Symbol

Question 4

- (a) Parse the following sentences to determine whether they are valid in Grammar A. If the sentence is invalid explain why.
- (i) $\times \nabla \times$
 - (ii) $\times \times \times \nabla \nabla \times \times$
 - (iii) $\times \times \times \times \nabla O \nabla \times \times$
 - (iv) $OOO \nabla \times \times \nabla O$
- (b) Determine whether the following sentences can be parsed by Grammar A, Grammar B, both or neither.
- (i) $OO \nabla OO$
 - (ii) $\times \times \nabla \times \times \times \nabla$
 - (iii) $OO \nabla OOOO$

Question 5

- (a) Modify Grammar A so that it only allows a single \times or O and not groups of those symbols.

For example, $\times \nabla O \nabla \times$ would be allowed but $\times O \nabla O \times \nabla \times \times$ would not.

- (b) What would be the effect on the sentences produced by Grammar B if the rule for S was removed and R became the start symbol?

Grammar B would become:

$$\begin{aligned} R &::= D \nabla D \mid D \\ D &::= DD \mid DDD \\ D &::= \times \mid O \end{aligned}$$

Use examples in your explanation.

Question 6

- (a) Convert the original Grammar A to a Railway Syntax Diagram.

- (b) ‘One grammar is a subset of the other.’

Explain the meaning of the above statement with reference to original Grammar A and B. Use examples in your explanation.

Section C: Criterion 8

Question 7

- (a) An electronic alarm system in a business operates using a number of sensors placed around the building. The alarm will sound if the alarm is armed and there is noise in the building or if lights are on while the doors are locked.

The alarm systems program uses the following boolean variables:

A = System is armed
 N = Noise in the building
 L = Lights off
 D = Doors locked
 S = Sound Alarm

Give an expression for the boolean variable S.

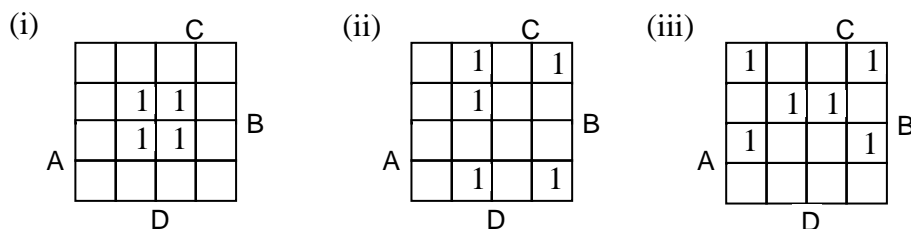
- (b) Simplify the following boolean expressions stating the logic laws used:
- (i) not x and x
 - (ii) (y and y) and (y or y)
 - (iii) (y and x) or (x and y)

Question 8

- (a) Evaluate the following Pascal expression if $x > y$:

- (i) $x \geq y$
- (ii) $(x < y) \text{ and } (y \leq x)$
- (iii) $(\text{not}(x \diamond x) \text{ and } (x > y)) \text{ or } (x = 6)$

- (b) Create a simple boolean expression for each of the following Karnaugh maps:



Question 9

(a) Using the logic laws show that the expression

$$\text{not (not (Y or (X and Y)) and (not X or X))}$$

is equivalent to the expression

$$Y$$

(b) Operator @ has three operands and is defined by

	A	
	1	1
		1
B		

A	
B @	(~B and ~C) or (~A and B and C)
C	

Give the Karnaugh map for the following expressions:

(i)
$$\begin{matrix} & B \\ C @ & \\ & A \end{matrix}$$

(ii)
$$\begin{matrix} & A & & B \\ B @ & \text{and} & C @ & \\ & C & & A \end{matrix}$$

(iii)
$$\begin{matrix} & A & & B \\ B @ & \text{or} & C @ & \\ & C & & A \end{matrix}$$

Section D: Criterion 10

Question 10

- (a) Using a word size of 12 bits and 2's complement representation, carry out the following calculations:
- (i) $0001\ 1100\ 1010 + 0000\ 1100\ 1001$
 - (ii) $0001\ 1000\ 1101 - 1111\ 0011\ 0101$
- (b) Give the value represented by the 6 bit word 111100 in the following representations:
- (i) 1's complement
 - (ii) 2's complement
 - (iii) Sign-magnitude

Question 11

- (a) If a computer has a word size is 16 bits, how many bits would be needed to store the following record type. Carefully explain your answer.

```
Recordtype = record
    A, B: char;
    C: packed array[ boolean] of char;
    D: integer
end
```

- (b) (i) Character representations normally give letters of the alphabet consecutive ordinal values. How does this help programs to sort text in alphabetical order?
- (ii) Explain how the following code converts letters from upper-case to lower-case:

```
chr(ord('a') + (ord(Letter) - ord('A')))
```

Question 12

- (a) A set of 'A' .. 'Z' could be represented using *array['A' .. 'Z'] of boolean*. Using this array structure explain how the operation of set union would operate.
- (b) A 16 bit floating-point representation has an eight bit mantissa followed by a seven bit exponent and then a sign bit. The exponent uses biased exponent representation with an excess of 63. Give the bit pattern for -9.5 using this representation.

BLANK PAGE

Section E: Criterion 11

All questions in this section refer to the SAM program **opposite**.

This program checks a three(3) digit PIN to determine if it is correct. It will encode the PIN to compare it with a stored set of two characters.

Question 13

- (a) Describe the purpose of the following instructions:
- (i) INC
 - (ii) RD
- (b) Assume the two locations *Digit* and *Sixtyfour* contain the following data:

Digit	0000 0000 0000 1010
Sixtyfour	0000 0000 0100 0000

Given this data, what would be the result of executing the following code?

LD	Digit
ADD	Sixtyfour
LS	8

Question 14

- (a) What validation check is carried out in lines 1 to 9?
- (b) Trace lines 13 to 16 noting that the initial value of *Check1* is zero. What will happen the next time this code is executed? What is the purpose of the variable *Check1*?
- (c) Trace the code from lines 27 to 35 assuming the value of *Digit* is 1, *Digit1* is 3 and *Digit2* is 6. Describe the method used to encode the PIN.

Question 15

- (a) The Pascal code given on the next page carries out the same process as the SAM code from lines 1 to 26. Indicate any differences between the method used by the SAM code and its equivalent Pascal code.
- (b) What is a correct PIN for this program? Explain your answer.

SAM Program

	<i>Address</i>	<i>Operation</i>	<i>Operand</i>
1	ENTER	LD	Prompt
2		WR	
3		RD	
4		ST	Digit
5		SUB	Zero
6		BLZ	ENTER
7		LD	Digit
8		SUB	Nine
9		BGZ	ENTER
10		LD	Digit
11		AND	Mask
12		ST	Digit
13		LD	Check1
14		BGZ	SECOND
15		INC	
16		ST	Check1
17		LD	Digit
18		ST	Digit1
19		B	ENTER
20	SECOND	LD	Check2
21		BGZ	LAST
22		INC	
23		ST	Check2
24		LD	Digit
25		ST	Digit2
26		B	ENTER
27	LAST	LD	Digit
28		ADD	Digit2
29		ADD	Sixtyfour
30		ST	Digit2
31		LD	Digit
32		ADD	Digit1
33		ADD	Sixtyfour
34		LS	8
35		ADD	Digit2
36		SUB	Code
37		BZ	CORRECT
38		LD	Answer
39		INC	
40		ST	Answer
41	CORRECT	H	

Data

<i>Address</i>	<i>Contents</i>	
Digit	0000 0000 0000 0000	
Digit1	0000 0000 0000 0000	
Digit2	0000 0000 0000 0000	
Check1	0000 0000 0000 0000	
Check2	0000 0000 0000 0000	
Code	0100 1010 0100 1101	ASCII JM
Answer	0000 0000 0000 0000	
Mask	0000 0000 0000 1111	
Prompt	0000 0000 0011 1110	ASCII >
Zero	0000 0000 0011 0000	ASCII 0
Nine	0000 0000 0011 1001	ASCII 9
Sixtyfour	0000 0000 0100 0000	

Pascal Program

```

read(Digit);
while not (Digit in [0 .. 9]) do read(Digit);
while not (Check1 and Check2) do
begin
  if not Check1 then
  begin
    Digit1 := Digit;
    Check1 := true;
  end
  else
    if not Check2 then
    begin
      Digit2 := Digit;
      Check2 := true;
    end;
  read(Digit);
  while not (Digit in [0 .. 9]) do read(Digit);
end;

```

IF862 COMPUTER SCIENCE

Official List of Logic Laws

Commutative Laws

L1	a	b	b	a
L2	a	b	b	a
L3	$(a$	$b)$	$(b$	$a)$

Associative Laws

L4	a	$(b$	$c)$	$(a$	$b)$	c
L5	a	$(b$	$c)$	$(a$	$b)$	c

Law of Negation

L6	$\sim \sim a$	a
----	---------------	-----

Distribution Laws

L7	a	$(b$	$c)$	$(a$	$b)$	$(a$	$c)$
L8	a	$(b$	$c)$	$(a$	$b)$	$(a$	$c)$
L9	a	$(b$	$c)$	$(a$	$b)$	$(a$	$c)$

De Morgan's Laws

L10	$\sim(a$	$b)$	$\sim a$	$\sim b$
L11	$\sim(a$	$b)$	$\sim a$	$\sim b$

Idempotent Laws

L12	a	a	a
L13	a	a	a

Law of the Excluded Middle

L14	a	$\sim a$	T
-----	-----	----------	-----

Law of Contradiction

L15	a	$\sim a$	F
-----	-----	----------	-----

Laws of Implication

L16	a	b	$\sim a$	b	
L17	a	b	$(a$	b	$b)$
L18	a	b	$(a$	b	$a)$
L19	a	b	a		
L20	a	a	b		

Laws of the Constants

L21	$\sim T$	F	
L22	$\sim F$	T	
L23	a	T	a
L24	a	F	F
L25	a	T	T
L26	a	F	a

Complement Rules of and

L27	a	$(b$	$\sim a)$	a	b
L28	a	$(b$	$\sim a)$	a	b

Equivalences Laws

L29	$\sim(a$	$b)$	$(\sim a$	$b)$		
L30	$\sim(a$	$b)$	$(a$	$\sim b)$		
L31	$(a$	$b)$	$(\sim a$	$b)$	$(a$	$\sim b)$
L32	$(a$	$b)$	$(a$	$b)$	$(b$	$a)$

Absorption Laws

L33	a	$(a$	$b)$	a
L34	a	$(a$	$b)$	a

Monotonicity Laws

L35	$(a$	$b)$	$(a$	c	b	$c)$
L36	$(a$	$b)$	$(a$	c	b	$c)$