Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

BACHELOR OF SCIENCE (HONS) IN
- APPLIED COMPUTING
- COMPUTER FORENSICS & SECURITY

EXAMINATION:

DISCRETE MATHEMATICS (COMMON MODULE)

SEMESTER 1 - YEAR 1

DECEMBER 2024 **DURATION: 2 HOURS**

INTERNAL EXAMINERS:	**DR DENIS FLYNN**	**DATE:**	**16 DEC 2024**
	DR KIERAN MURPHY	**TIME:**	**11.45 AM**
		VENUE:	**MAIN HALL**
EXTERNAL EXAMINER:	**DR JULIE CROWLEY**		

INSTRUCTIONS TO CANDIDATES

1. **ANSWER ALL QUESTIONS.**
2. **TOTAL MARKS = 100.**
3. **EXAM PAPER (5 PAGES EXCLUDING THIS COVER PAGE), FORMULA AND PYTHON SHEETS (2 PAGES)**

MATERIALS REQUIRED

1. **NEW MATHEMATICS TABLES.**
2. **GRAPH PAPER**

SOUTH EAST TECHNOLOGICAL UNIVERSITY

Question 1

(a) Convert the following English sentence into a propositional logic expression with **atomic** propositions.

> "*If the internet connection is stable and the server is running, then the website will be accessible.*"

(2 marks)

(b) Let R be a relation on the set $A = \{1, 2, 3\}$, where $R = \{(1, 1), (2, 2), (3, 3), (1, 2)\}$.

(i) Is R reflexive? symmetric? transitive? Justify your answer.

(ii) Is R an equivalence relation? Justify your answer.

(iii) Is R antisymmetric? Justify your answer.

([3×2] **6 marks)**

(c) How many 12-bit strings (that is, bit strings of length 12) exist which satisfy each of the following criteria? (Justify your answers.)

(i) Start with the sub-string 1101.

(ii) Have weight 6 (i.e., contain exactly six 1's) and start with the sub-string 1101.

(iii) Either start with 1101 or end with 1010 (or both).

(iv) Have weight 6, start with 1101, and end with 1010.

([$1 + 1 + 2 + 2$] **6 marks)**

(d) Use a truth table to determine whether the proposition

$$(\neg P \vee Q) \rightarrow (P \wedge \neg R) \vee (Q \wedge R)$$

is a contradiction, is satisfiable, or is a tautology. **(6 marks)**

(Total 20 marks)

Question 2

(a) A firewall follows the following rules to allow or block network traffic. Translate each of the firewall rules into a propositional logic expression, using the following atomic propositions:

- $T =$ "*The IP address is trusted.*"
- $S =$ "*The protocol is secure.*"
- $A =$ "*The connection is allowed.*"

(i) If a connection request is from a trusted IP address and uses a secure protocol, the connection will be allowed.

(ii) If the connection is from a suspicious IP address or uses an insecure protocol, the connection will be blocked.

(iii) A connection from a trusted IP address will always be allowed, even if it uses an insecure protocol.

(iv) A suspicious IP address will always be blocked, regardless of the protocol used.

(8 marks)

(b) Consider lattice paths on a grid from $(0, 0)$ to $(7, 4)$, and you can only move right or up.

(i) How many distinct paths are there from $(0, 0)$ to $(7, 4)$?

(ii) How many paths pass through $(4, 2)$?

(iii) How many paths pass through either $(3, 1)$ or $(5, 3)$?

($[2 + 3 + 4]$ **9 marks)**

(c) The first term of a geometric progression is 10, and the common ratio is 0.5. Determine whether the number 0.5 is a term in this progression. If yes, then determine the position of 0.5 in the sequence. **(3 marks)**

(Total 20 marks)

Question 3

(a) Let $f : \mathbb{R} \to \mathbb{R}$ be defined by $f(x) = 3x - 5$.

(i) Explain why f has a inverse.

(ii) Determine $f^{-1}(x)$.

(iii) Verify that $f^{-1}(f(x)) = x$ and $f(f^{-1}(x)) = x$.

([1 + 2 + 2] 5 marks)

(b) A digital image starts at a resolution of 1024×768 pixels. We need to reduce the resolution by half for each step (in both dimensions) until the image size reaches a required value.

(i) Express the **number of pixels** at each step as a geometric sequence.

(ii) Find the total number of pixels after three reduction steps.

(iii) How many reductions are needed until the number of pixels drops below 1,000?

([2 + 2 + 3] 7 marks)

(c) Consider the following python code. Translate to mathematical notation using summation ($\sum$) and/or product ($\prod$) notation. Hence, or otherwise, determine the output generated.

```
var_a = 0
for k in range(1,3):
    var_b = 1
    for j in range(k):
        var_b *= (j+k)**2
    var_a += var_b

print(f"{var_a = }")
```

(5 marks)

(d) The first term of an arithmetic progression is 2, and the common difference is 4. What is the 10th term of this arithmetic progression? **(3 marks)**

(Total 20 marks)

Question 4

(a) Let $A = \{0, 1, 2, 3, 4\}$, $B = \{3, 4, 5, 6\}$, and $C = \{1, 4, 5, 6, 7\}$, and let the universal set be $U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Determine each of the following.

(i) $A \cup B$ **(iii)** $B \cap A$ **(v)** $(A \cup B) \cap C$

(ii) $A \setminus B$ **(iv)** $A \oplus B$ **(vi)** $\bar{A}$

([6×1] 6 marks)

(b) Let $S = \{2, 3, 5, 7, 11, 13, 17\}$.

(i) How many subsets are there of cardinality 3?

(ii) How many subsets of S are there? That is, find $|\mathcal{P}(S)|$.

(iii) How many subsets of S are there where the sum of the elements equals 15?

([$1 + 1 + 2$] 4 marks)

(c) Design a logical circuit that represents the logical expression,

$$(A \wedge (B \vee C)) \vee (\neg B \wedge \neg C).$$

(5 marks)

(d) What does the following function compute? (Justify your answer.)

```
def isWhat(A, B):
    for a in A:
        if a not in B: return False
    return True
```

(5 marks)

(Total 20 marks)

Question 5

(a) Draw a graph with degree sequence $(3, 3, 5, 5, 5, 5)$. Does there exist a *simple* graph with this degree sequence? Justify your answer. **(2 marks)**

(b) Consider an employee database system for a company storing *Employees* and *Departments*. A developer has coded a function

$$f(\text{employee})$$

which returns the department in which the employee works.

(i) Can this function be injective (one-to-one)? Justify your answer

(ii) Can this function be surjective (onto)? Justify your answer.

(4 marks)

(c) Consider the functions defined by the following Python code:

```
def f(x):
    return 2*x + 3

def g(x):
    return x**3 - 1

def h(x):
    return 5 / x
```

Evaluate the following:

(i) $f(4)$ **(ii)** $g(2)$ **(iii)** $f(5) + g(2) - h(4)$

(iv) $f(g(1))$ **(v)** $g(h(2))$ **(vi)** $h(f(3))$

([6 × 1] 6 marks)

(d) The **girth** of a graph is the length of its shortest cycle. Write down the girth of each of the following graphs.

(i) K_9

(ii) $K_{5,7}$

(iii) C_8

(iv) W_8

(8 marks)

(Total 20 marks)

Laws of Logic

Logical Connective	Symbol	Python Operator	Precedence	Logic Gate
Negation (NOT)	$\neg$	**not**	Highest	
Conjunctive (AND)	$\wedge$	**and**	Medium	
Disjunctive (OR)	$\vee$	**or**	Lowest	

Basic Rules of Logic

Commutative Laws

$$p \vee q \Leftrightarrow q \vee p \qquad p \wedge q \Leftrightarrow q \wedge p$$

Associative Laws

$$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r) \qquad (p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$$

Distributive Laws

$$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r) \qquad p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$$

Identity Laws

$$p \vee \mathbf{F} \Leftrightarrow p \qquad p \wedge \mathbf{T} \Leftrightarrow p$$

Negation Laws

$$p \wedge (\neg p) \Leftrightarrow \mathbf{F} \qquad p \vee (\neg p) \Leftrightarrow \mathbf{T}$$

Idempotent Laws

$$p \vee p \Leftrightarrow p \qquad p \wedge p \Leftrightarrow p$$

Null Laws

$$p \wedge \mathbf{F} \Leftrightarrow \mathbf{F} \qquad p \vee \mathbf{T} \Leftrightarrow \mathbf{T}$$

Absorption Laws

$$p \wedge (p \vee q) \Leftrightarrow p \qquad p \vee (p \wedge q) \Leftrightarrow p$$

DeMorgan's Laws

$$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q \qquad \neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$$

Involution Law

$$\neg(\neg p) \Leftrightarrow p$$

Implications and Equivalences

Detachment (Modus Ponens)

$$(p \to q) \wedge p \Rightarrow q$$

Indirect Reasoning (Modus Tollens)

$$(p \to q) \wedge \neg q \Rightarrow \neg p$$

Disjunctive Addition

$$p \Rightarrow (p \vee q)$$

Conjunctive Simplification

$$(p \wedge q) \Rightarrow p \qquad (p \wedge q) \Rightarrow q$$

Disjunctive Simplification

$$(p \vee q) \wedge \neg p \Rightarrow q \qquad (p \vee q) \wedge \neg q \Rightarrow p$$

Chain Rule

$$(p \to q) \wedge (q \to r) \Rightarrow (p \to r)$$

Resolution

$$(\neg p \vee r) \wedge (p \vee q) \Rightarrow (q \vee r)$$

Conditional Equivalence

$$p \to q \Leftrightarrow \neg p \vee q$$

Biconditional Equivalences

$$(p \leftrightarrow q) \Leftrightarrow (p \to q) \wedge (q \to p)$$
$$\Leftrightarrow (p \wedge q) \vee (\neg q \wedge \neg q)$$

Contrapositive

$$p \to q \Leftrightarrow \neg q \to \neg p$$

Python Cheat Sheet

Base Types

integer, float, boolean, string, bytes

int 163 0 −192 0b110 (binary) 0x3F (hex)

float 9.32 0.0 −1.7E−6 (×10⁻⁶)

bool False (0) True (1)

str 'some text' or "some text"

bytes b"text\xfe\775"

Container Types

ordered containers — repeatable values

list	[1,5,3]	["a",1,5,5]	[5]	[]
tuple	(1,5,3)	"a",1,5,5	(5,)	()

Immutable (non-modifiable values)

str	"153"	""

key containers — no order, unique keys

set	{"key1","key2"}	{1,9,3,0}	**set**()
dict	{"key1":value1,"key2":value2}	**dict**(a=3,b="v")	{}

Integer Sequences

range([start,] end [,step])

start default is 0 (inclusive), end (exclusive), step default is 1.

range(5) → 0,1,2,3,4
range(2,5) → 2,3,4
range(2,12,3) → 2,5,8,11
range(20,5,−5) → 20,15,10

Conversions

type(expression)

int('153') → 15
int('3f',16) → 63 (Specify base in 2nd parameter)
int(−11.24e8) → −1124000000
int(15.56) → 15 (Truncate decimal point)
round(15.58,1) → 15.6 (Round to 1 decimal place)
float('15.56') → 15.56

bool(x) (False for None, zero or empty containers)
str(x) (String representation of x.)

chr(65) → 'A' code ↔ char **ord**('A') → 65

list('abc') → ['a','b','c']
dict([(3,'three'), (1,'one')]) → {3:'three', 1:'one'}
set(['one','two']) → {'one','two'}

(Split string using a separator, **str** → **list** of **str**)
'random:data:666'.split(':') → ['random', 'data', '666']

(Join a list of strings, **list** of **str** → **str**)
':'.join(['random', 'data', '666']) → 'random:data:666'

(Convert each element in a collection)
[**int**(x) **for** x **in** ['1','29','−3']] → [1, 29, −3]

Operations on Sets

Operators

|	.union
&	.intersection
−	.difference

Methods

s.add(key)	s.update(s2)
s.clear()	s.remove(key)

Operations on Lists

Methods

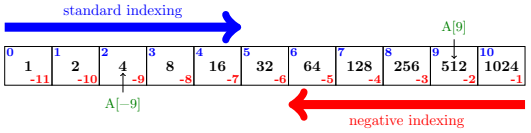
a.append(value)	a.extend(a2)
s.insert(idx,value)	a.pop()

Generic Operations on Containers

min(c) **max**(c) **sum**(c) **sorted**(c)
len(c) (Number of elements in collection c)

all(c) → True if **all** items in c evaluate to True, else False.
any(c) → True if **at least one** item in c evaluate to True, else False.

Sequence Containers Indexing



a[3:6] → [8, 16, 32]
a[1:−1] → [2, 4, 8, 16, 32, 64, 128, 256, 512]
a[::−1] → [1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1]

Looping over Collections

(Loop over values)
for value **in** A:
  print(value)

(Count and loop over values)
for k,value **in** **enumerate**(A):
  print(k, value)

(While loop)
k = 0
while k<**len**(A):
  print(k, A[k])
  k += 1

Initialisation **before** loop.
update **within** loop.

break immediatly exits loop. **continue** skips to next iteration.
else block for normal loop exit.