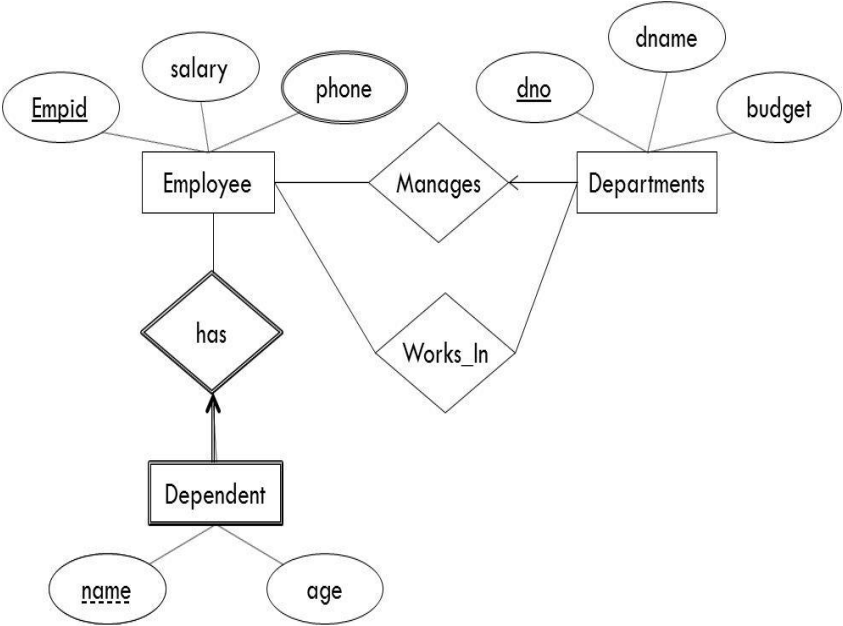
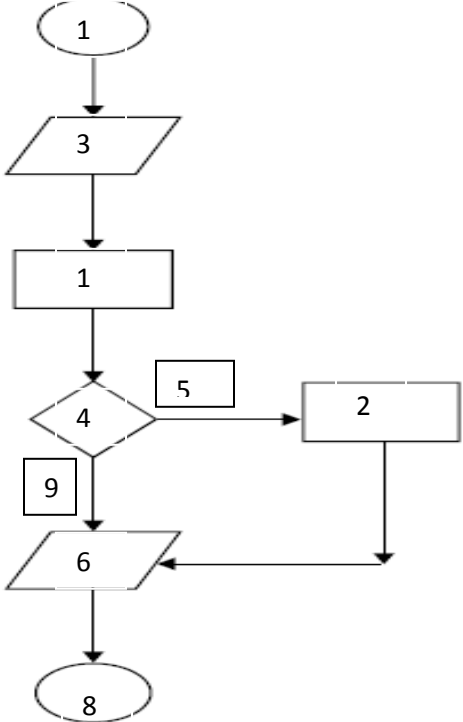


ICT -2016 (Grade 12) July – Third Term Test- MCQ Paper
Marking Scheme

(1)	2	(11)	3	(21)	3	(31)	1	(41)	3
(2)	3	(12)	3	(22)	1	(32)	4	(42)	2
(3)	5	(13)	1	(23)	2	(33)	3	(43)	1
(4)	4	(14)	5	(24)	4	(34)	4	(44)	4
(5)	1	(15)	2	(25)	5	(35)	3	(45)	3
(6)	1	(16)	2	(26)	1	(36)	1	(46)	2
(7)	5	(17)	5	(27)	4	(37)	2	(47)	5
(8)	3	(18)	1	(28)	3	(38)	4	(48)	3
(9)	4	(19)	4	(29)	5	(39)	2	(49)	1
(10)	2	(20)	4	(30)	2	(40)	5	(50)	5

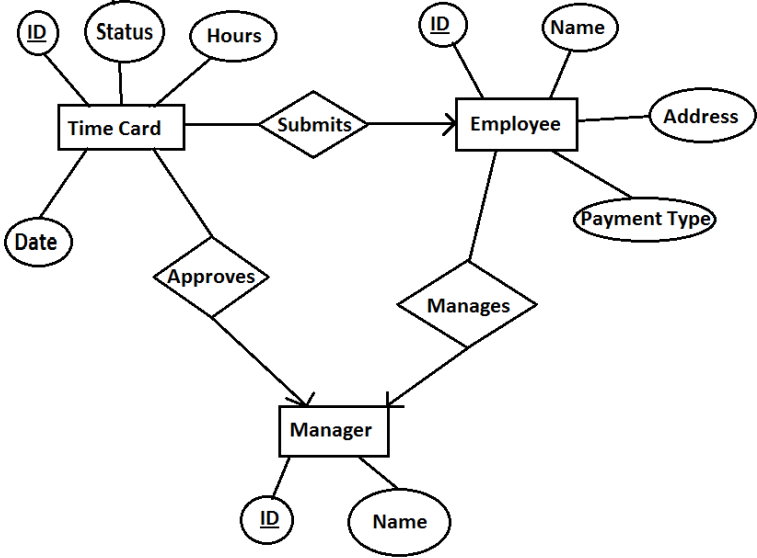
ICT -2016 (Grade 12) July – Third Term Test-Part A (Structured Essay)

Question No	Suggested answers	Marks
1.(a)	$ \begin{array}{r} 51 \\ \text{a) } 24 \overline{) 1464} \\ \underline{144} \\ 24 \\ \underline{24} \\ 0 \end{array} $	4
1.(b)	computer with cache memory because speed of a computer with cache memory is faster than a computer without cache memory	3
1.(c)	$F(A,B,C) = A'B+ABC$ <i>standard SOP expression</i> = $A'BC+A'BC'+ABC$	3
2.(a)	<i>list1[0]: ICT</i> <i>list2[1:5]: [2, 3, 4, 5]</i>	3
2.(b)	<i>The count is: 0</i> <i>The count is: 1</i> <i>The count is: 2</i> <i>The count is: 3</i> <i>Good bye!</i>	3
2.(c)	<i>count = 0</i> <i>for i in range(0,5):</i> <i> print i,"is less than 5"</i> <i> print i+1,"is not less than 5"</i>	4
3.(a)	<i>Entities are : employees,department, child</i> <i>Relationships: dependent(has),works in,manages</i>	3

Question No	Suggested answers	Marks
3.(b)	 <p>The diagram shows three entities: Employee, Dependent, and Departments. Employee has attributes Empid (underlined), salary, and phone. Dependent has attributes name (underlined) and age. Departments has attributes dno (underlined), dname, and budget. Relationships: 'Manages' connects Employee and Departments; 'Works_In' connects Employee and Departments; 'has' connects Employee and Dependent. An arrow points from Dependent to 'has'.</p>	4
3.(c)	<p><i>One to one: primary key of table1 insert to table2 as foreign key or primary key of table2 use as foreign key of table1</i></p> <p>One to many:</p> <p>Many to many :</p>	3
4.(a)	 <p>The flowchart starts with oval 1, leading to parallelogram 3, then rectangle 1, then diamond 4. From diamond 4, one path goes to rectangle 2, and another path goes to parallelogram 6. From rectangle 2, an arrow points to parallelogram 6. From parallelogram 6, an arrow points to oval 8. There are also small boxes with numbers 5 and 9 near diamond 4.</p>	03

Question No	Suggested answers	Marks
(b)	<p><i>1.testing</i> <i>2.Analysis</i> <i>3.Implement</i> <i>4.Development</i></p>	03
(c)	<p><i>One's complement</i> 8 0000 1000 <u>-5 1111 1010</u> 10000 0010 _____ 1 0000 0011</p> <p><i>Two's complement</i> 8 0000 1000 <u>-5 1111 1011</u> 10000 0011 ↙ neglect</p>	04

ICT -2016 (Grade 12) July – Third Term Test-Part B (Essay)
Marking Scheme

Questi on No	Suggested answers	Marks																						
1.	 <pre> erDiagram TimeCard --o{ Employee : Submits TimeCard --o{ Manager : Approves Employee --o{ Manager : Manages TimeCard { string ID PK string Status float Hours string Date } Employee { string ID PK string Name string Address string PaymentType } Manager { string ID PK string Name } </pre>	25																						
2	<p>Python String literals can be enclosed in matching single quotes (') or double quotes ("). They can also be enclosed in matching groups of three single or double quotes (these are generally referred to as <i>triple-quoted strings</i>). The backslash (\) character is used to escape characters that otherwise have a special meaning, such as newline, backslash itself, or the quote character. String literals may optionally be prefixed with a letter 'r' or 'R'; such strings are called raw strings and use different rules for backslash escape sequences</p> <p>Example:</p> <table border="1" data-bbox="480 1400 1146 1864"> <thead> <tr> <th>Escape Sequence</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td><code>\newline</code></td> <td>Ignored</td> </tr> <tr> <td><code>\\</code></td> <td>Backslash (\)</td> </tr> <tr> <td><code>\'</code></td> <td>Single quote (')</td> </tr> <tr> <td><code>\"</code></td> <td>Double quote (")</td> </tr> <tr> <td><code>\a</code></td> <td>ASCII Bell (BEL)</td> </tr> <tr> <td><code>\b</code></td> <td>ASCII Backspace (BS)</td> </tr> <tr> <td><code>\f</code></td> <td>ASCII Formfeed (FF)</td> </tr> <tr> <td><code>\n</code></td> <td>ASCII Linefeed (LF)</td> </tr> <tr> <td><code>\r</code></td> <td>ASCII Carriage Return (CR)</td> </tr> <tr> <td><code>\t</code></td> <td>ASCII Horizontal Tab (TAB)</td> </tr> </tbody> </table>	Escape Sequence	Meaning	<code>\newline</code>	Ignored	<code>\\</code>	Backslash (\)	<code>\'</code>	Single quote (')	<code>\"</code>	Double quote (")	<code>\a</code>	ASCII Bell (BEL)	<code>\b</code>	ASCII Backspace (BS)	<code>\f</code>	ASCII Formfeed (FF)	<code>\n</code>	ASCII Linefeed (LF)	<code>\r</code>	ASCII Carriage Return (CR)	<code>\t</code>	ASCII Horizontal Tab (TAB)	
Escape Sequence	Meaning																							
<code>\newline</code>	Ignored																							
<code>\\</code>	Backslash (\)																							
<code>\'</code>	Single quote (')																							
<code>\"</code>	Double quote (")																							
<code>\a</code>	ASCII Bell (BEL)																							
<code>\b</code>	ASCII Backspace (BS)																							
<code>\f</code>	ASCII Formfeed (FF)																							
<code>\n</code>	ASCII Linefeed (LF)																							
<code>\r</code>	ASCII Carriage Return (CR)																							
<code>\t</code>	ASCII Horizontal Tab (TAB)																							

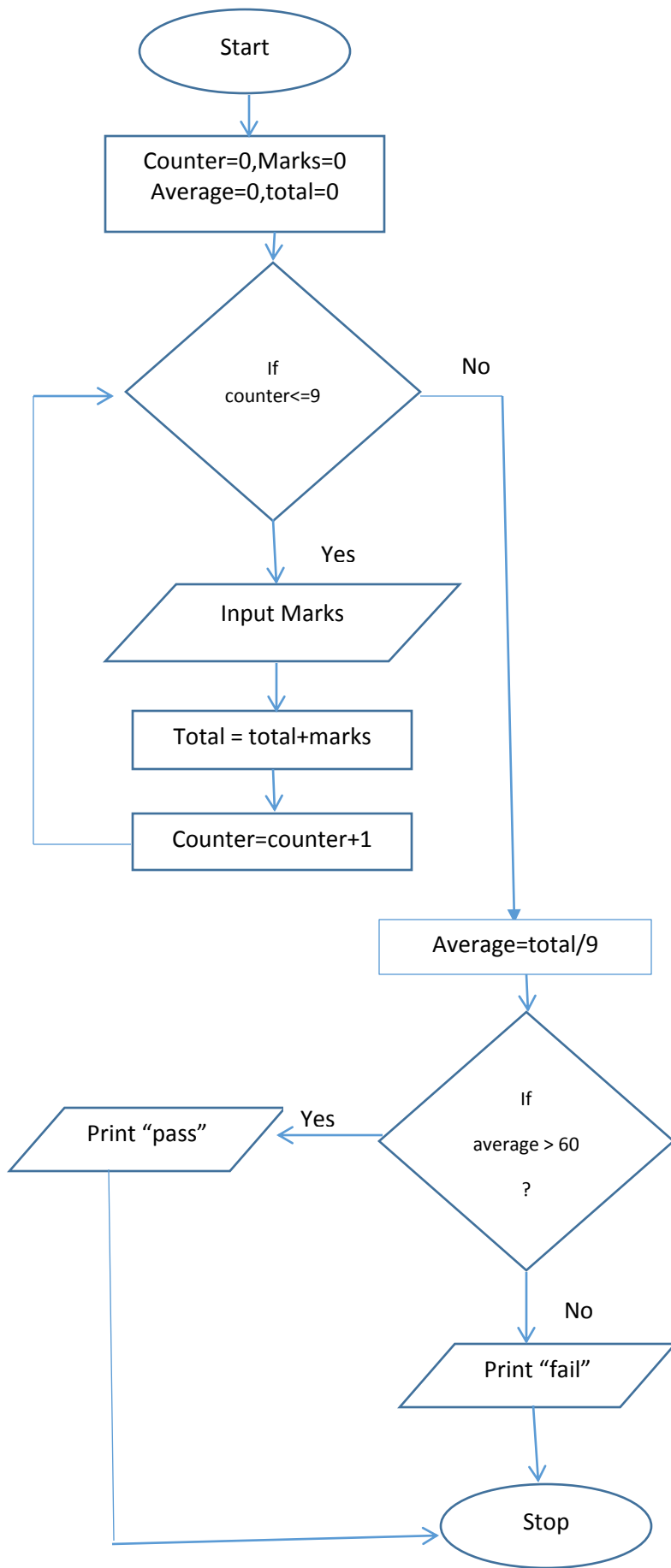
Indentation:

Whitespace is important in Python. Actually, **whitespace at the beginning of the line is important**. This is called **indentation**. Leading whitespace (spaces and tabs) at the beginning of the logical line is used to determine the indentation level of the logical line, which in turn is used to determine the grouping of statements.

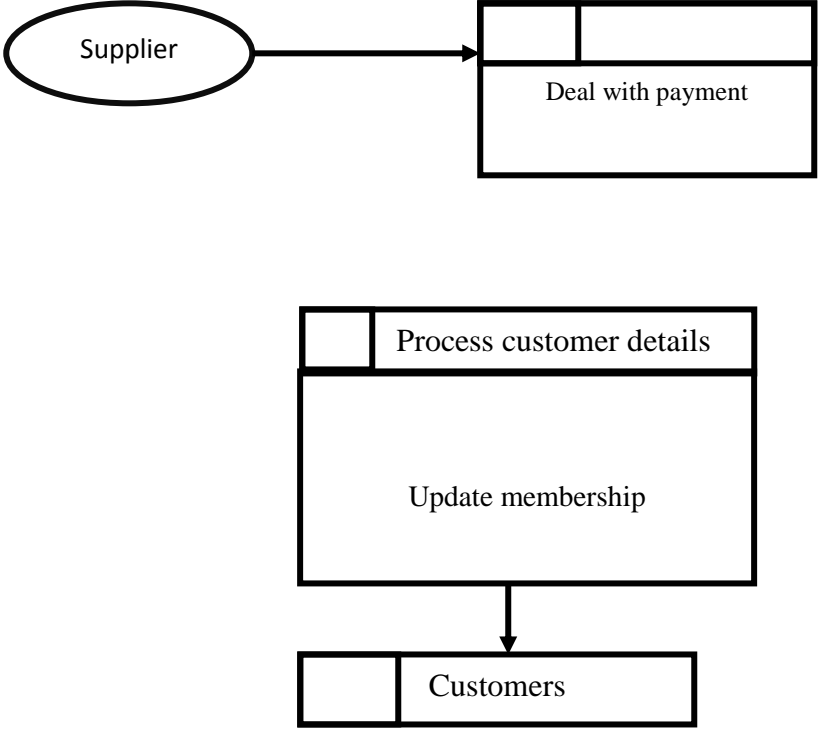
This means that statements which go together **must** have the same indentation. Each such set of statements is called a **block**.

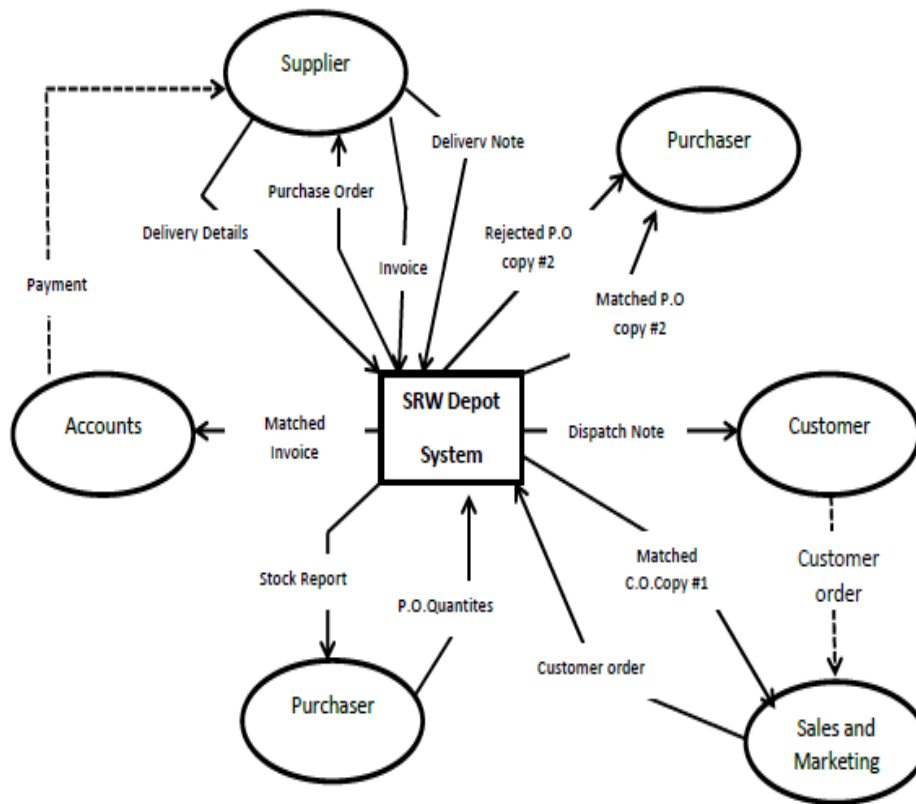
Example:

```
s = 1
for x in range(1, 5):
    s = s + x (whitespace at the beginning of this line is called indentation)
    print s
```



	<p><u>Pseudo code</u> Begin Counter=0,marks=0,average=0 Total =0 While counter<=9 Input marks Total=total+marks Counter=counter+1 End while Average=total/9 If average>60 then Print “pass” Else Print “fail” Endif End.</p> <p><u>Python code</u></p> <pre> counter =0 marks=0 total=0 avegrage=0 while(counter<=9): marks=int(input('enter marks')) total=total+marks counter+=1 average=total/9 if average > 60: print('pass') else: print('fail') </pre>	15
3(a)	<ul style="list-style-type: none"> - <u>Transaction Processing Systems (TPS)</u>: are the basic business systems that serve the operational level of the organization. And it is also a computerized system that performs and records the daily routine transactions necessary to conduct business. - Management Information Systems (MIS): serve the management level of the organization, providing managers with reports and often-online access to the organization’s current performance and historical records and primarily serve the functions of planning, controlling, and decision-making. - <u>Decision-Support System (DSS)</u>: also serve the management level or the organization. DSS help mangers make decisions that are unique, rapidly changing, and not easily specified in advance. - <u>Executive Support System (ESS)</u>: serve the strategic level of the organization. They address non routine decisions requiring judgment, evaluation, and insight because there is no agreed on procedure for arriving at a solution. 	5

<p>3(b)</p>	<p>The highest-level view of an organizational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system All context diagrams have only one process labeled “0” No data stores appear on a context diagram.</p> <p>Context diagrams, (a) describes the context of a system (b) is a DFD which gives an overview of the system (c) is a detailed description of a system</p> <p>A context diagram is used (a) as the first step in developing a detailed DFD of a system (b) in systems analysis of very complex systems (c) as an aid to system design (d) as an aid to programmers</p> <p>A context diagram is not used (a) Is not used in drawing a detailed DFD. (b) Is not used in drawing a document flow diagrams (c) No Data stores appear on a context diagram.</p>	<p>5</p>
<p>3(c)</p>	 <p>The first diagram shows an external entity 'Supplier' (represented by an oval) connected by an arrow to a process box labeled 'Deal with payment'. The process box has a small empty box in its top-left corner.</p> <p>The second diagram shows a process box labeled 'Update membership' connected by an arrow to an external entity 'Customers' (represented by a rectangle). The process box has a small empty box in its top-left corner.</p>	<p>5</p>



10

4(a)

A **minterm** is a Boolean expression resulting in 1 for the output of a single cell, and 0s for all other cells in a Karnaugh map, or truth table. If a **minterm** has a single 1 and the remaining cells as 0s, it would appear to cover a minimum area of 1s.

These terms are represent as follows:

1. if variable 0 it represent as complement of such variable
2. if variable 1 it represent as normal

maxterm (standard sum term) A sum (OR) of n Boolean variables, uncomplemented or complemented but not repeated, in a Boolean function of n variables. With n variables, 2^n different **maxterms** are possible. The complement of any **maxterm** is a minterm.

These terms are represent as follows:

1. if variable 0 it represent as normal
2. if variable 1 it represent as complement of such variable

5

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \bar{x}_1\bar{x}_2x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$
2	0	1	0	$m_2 = \bar{x}_1x_2\bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$
3	0	1	1	$m_3 = \bar{x}_1x_2x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	$m_4 = x_1\bar{x}_2\bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\bar{x}_2x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	$m_6 = x_1x_2\bar{x}_3$	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$

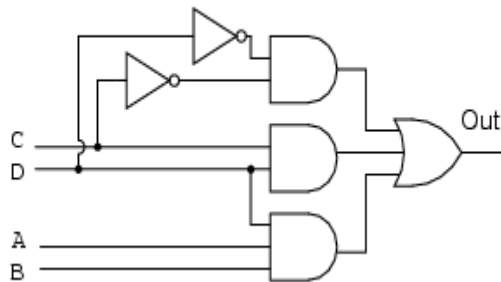
4(b)

$$\text{Out} = (A+B+C+\bar{D})(A+B+\bar{C}+D)(A+\bar{B}+C+\bar{D})(A+\bar{B}+\bar{C}+D) \\ (\bar{A}+\bar{B}+\bar{C}+D)(\bar{A}+B+C+\bar{D})(\bar{A}+B+\bar{C}+D)$$

$\bar{A} \backslash CD$	00	01	11	10
00		0		0
01		0		0
11				0
10		0		0

$\bar{A} \backslash CD$	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	1	1	1	0
10	1	0	1	0

$$\text{Out} = \bar{C}\bar{D} + CD + ABD$$



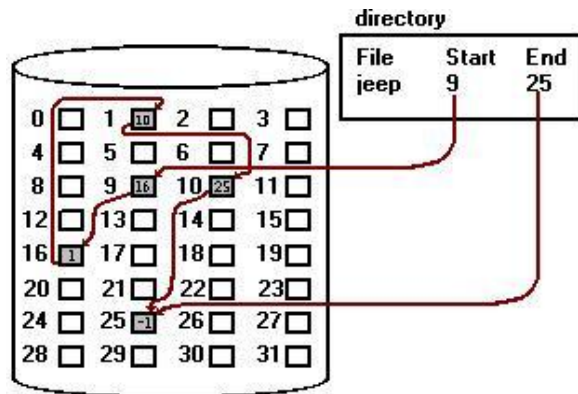
$\bar{A} \backslash CD$	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	1	1	1	0
10	1	0	1	0

$$\text{Out} = \bar{C}\bar{D} + CD + ABD$$

5(a)

The indexed allocation method is the solution to the problem of both contiguous and linked allocation. This is done by bringing all the pointers together into one location called the index block. Of course, the index block will occupy some space and thus could be considered as an overhead of the method. In indexed allocation, each file has its own index block, which is an array of disk sector addresses.

The i^{th} entry in the index block points to the i^{th} sector of the file. The directory contains the address of the index block of a file. To read the i^{th} sector of the file, the pointer in the i^{th} index block entry is read to find the desired sector. Indexed allocation supports direct access, without suffering from external fragmentation. Any free block anywhere on the disk may satisfy a request for more space.



5(b)	No	Event	Starting state	Ending State
	1.	Program started by user (පරිශීලක විසින් ක්‍රමලේඛය ආරම්භ කිරීම.)	Does not Exist	New
	2	Scheduler dispatches process (කෙටිකාලීන නියමකරණය මගින් ක්‍රියායතය ප්‍රවාහනය කිරීම.)	Ready	Running
	3	I/O complete (ආදාන / ප්‍රතිදාන සම්පූර්ණ කිරීම.)	Waiting	Ready
	4	Process admitted to ready queue for first time (ක්‍රියායතය මුල් වතාවට සුදානම් තත්වයේ පවතින ක්‍රියායත පෝලීමට එකතු වීම.)	New	Ready
	5	Scheduler preempts (interrupts) process (නියමකරකය මගින් ක්‍රියායතයට අතුරු බිඳුමක් ඇති කිරීම.)	Running	Ready
	6	Process finishes execution (task is complete) (ක්‍රියායතය ක්‍රියාත්මක වී අවසන් වීම.)	Running	Terminated
	7	Process initiates I/O (ක්‍රියායතය ආදාන / ප්‍රතිදාන ආරම්භ කිරීම.)	Running	Waiting