|  |  |
| --- | --- |
| COMP 2121 **DISCRETE MATHEMATICS** | Assignment 2  Fall 2024 |

|  |  |  |
| --- | --- | --- |
| Name: Vincent Feng A0??? | Set: 2D | |
| Name: Jiarui Xing A01354731 | Set: 2D | |
|  |  | |
| Section | Total | Actual |
| Question 1 | 10 |  |
| Question 2 | 10 |  |
| Question 3 | 15 |  |
| Question 4 | 10 |  |
| Question 5 | 15 |  |
| Total | 60 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\A00141222\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ARBZC6KC\MC900432530[1].png | | | Instructions |
|  |  |  | * Assignment must be done using Microsoft Word - type your work in this document. * Handwritten assignments will not be marked. * Create FSM drawings at <http://madebyevan.com/fsm/> then paste figures into this document. * FSM that is drawn by hand will not be marked. |
|  |  |  |
|  |  |  |
|  |  |  | * The assignment must be done in a **group** of two students – no individual assignments will be accepted. |
|  |  |  |
|  |  |  |
|  |  |  | * Be careful when writing sigma notation, do not forget to write upper, lower bound or similar. Marks will be taken off for missing details. |
|  |  |  |
|  |  |  |
|  |  |  | * When simplifying, show all algebra steps. Otherwise, marks will be taken away. |
|  |  |  |
|  |  |  |
|  |  |  | * **PRINT** the completed assignment – you are handing in a paper copy |
|  |  |  |
|  |  |  |
|  |  |  | * **Due** at the beginning of the **Lecture** on **Nov 6, 2024**. * No late assignments will be accepted. * Electronic copies will not be accepted. |
|  |  |  |
|  |  |  |

**Q1)** Given

Find

Write Final Answer here:

For **Q2)**, **Q3)** use the following formulas as necessary:

*\*\*\* In order to receive marks your approach must rely on sigma notation, summation properties and usage of given formula(s). \*\*\* Answers based on combinatorics will not be accepted.*

**Q2)** Consider the following programming segment:

time = 1000;

for (i = 1; i <= n; i++){

time = time + 9;

for (k = 5; k <= i+6; k++){

time = time + 12k

time = time + 4;

}

}

a) Find the value of the variable time (in terms of n) after the programming segment is executed. Your formula must be in the simplest form, fully expanded and terms combined.

(E.g. )

b) Test your formula by comparing the following numbers:

* Implement this code in the programming language of your choice. Use the value n = 50 and print the variable time after the code execution. You must provide the screenshots of implementation and output.
* Evaluate your formula from part a) for n = 50.
* What do you conclude?

**Q3)** Consider the following programming segment:

time = 1000;

for (i = 1; i <= n; i++){

time = time + 9;

for (k = 14; k <= i+6; k++){

time = time + 12k;

}

}

1. Find the value of the variable time (in terms of n) after the programming segment is executed. Your formula must be in the simplest form, fully expanded and terms combined.

(E.g. )

b) Test your formula by comparing the following numbers:

* Implement this code in the programming language of your choice. Use the value n = 50 and print the variable time after the code execution. You must provide the screenshots of implementation and output.
* Evaluate your formula from part a) for n = 50.
* What do you conclude?
* Use the link from above to draw FSM – this is not optional.
* FSM must be clearly labelled and nicely drawn – will not be marked otherwise.
* Make sure there are outgoing 0 and 1 from every state. If that is missing, the machine cannot be marked.
* FSM needs to have the minimum number of required states.
* Properly name the states; S0, S1, S2 or such are not acceptable.
* The start state must have an arrow pointing to it.
* Efforts must be made so that arcs do not intersect or have intersections reduced at the minimum. Mark penalty applies. It is easy to move the states around and check.
* Marks will be taken away for missing details.

**Q4**) Design a finite state machine with input alphabet {0, 1} that accepts all strings that have even length and number of 1’s not divisible by 4.

* Use the link from above to draw FSM – this is not optional.
* FSM must be clearly labelled and nicely drawn – will not be marked otherwise.
* Make sure there are outgoing 0 and 1 from every state. If that is missing, the machine cannot be marked.
* FSM needs to have the minimum number of required states.
* Properly name the states; S0, S1, S2 or such are not acceptable.
* The start state must have an arrow pointing to it.
* Efforts must be made so that arcs do not intersect or have intersections reduced at the minimum. Mark penalty applies. It is easy to move the states around and check.
* Marks will be taken away for missing details.

**Q5**) Design a finite state machine with input alphabet {0, 1} that accepts all strings that (start with 11 or end with 101) and contain an even number of 1’s.