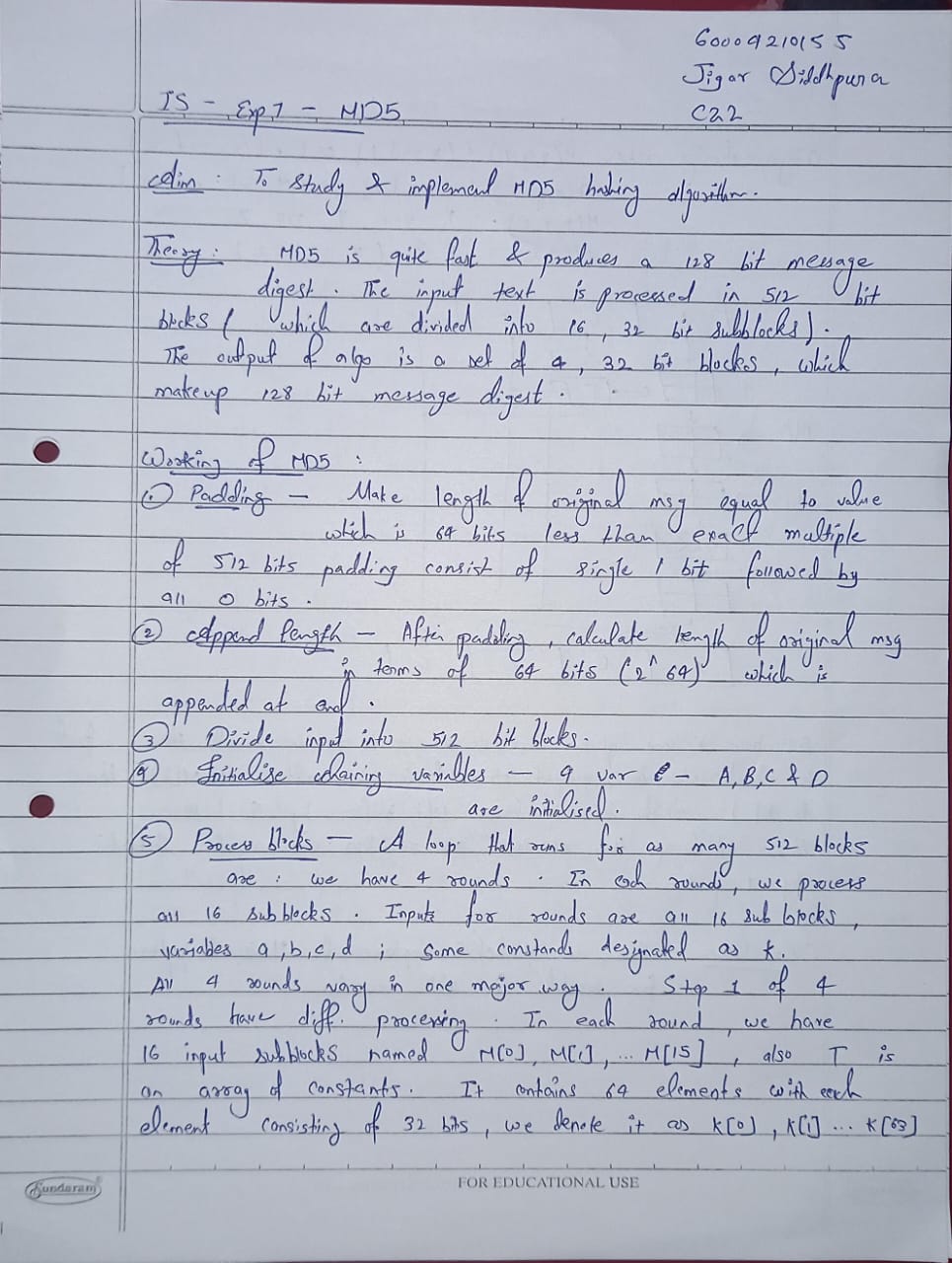
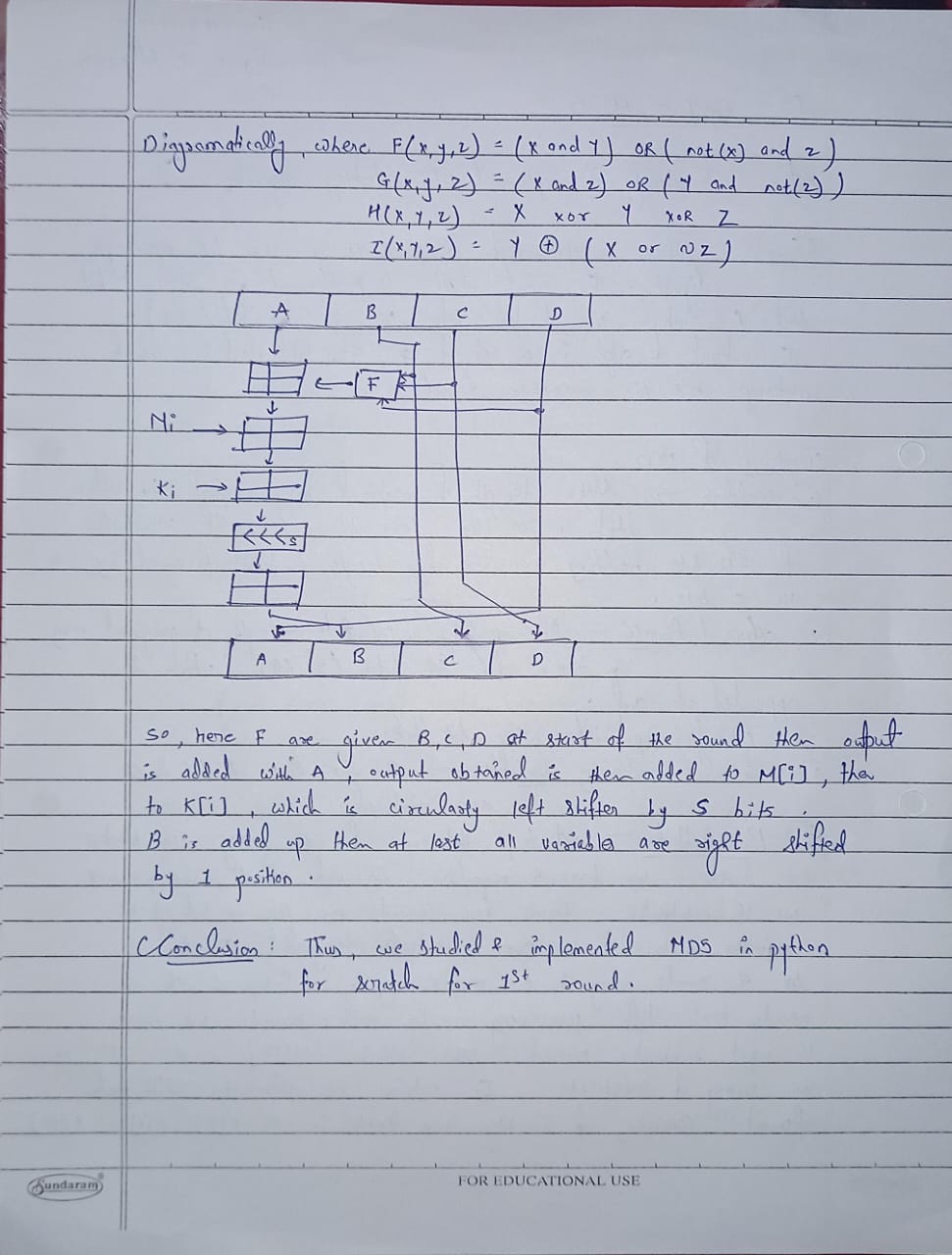
**Name: Jigar Siddhpura SAPID:** 60004200155

**DIV: C/C2 Branch:** Computer Engineering

IS - Experiment 7 - MD5 Algorithm





**CODE**

import random

def left\_rotate(x, n):

    return ((x << n) | (x >> (32 - n))) & 0xFFFFFFFF

def md5(message):

    """

    A function to calculate the MD5 hash of the input message.

    Takes a message as input and returns the MD5 hash in hexadecimal format.

    """

    # Step 1 and 2: Padding and Append Length

    padding\_length = 0

    if (len(message) + 64) % 512:

        message += "1"

        padding\_length = 512 - ((len(message) + 64) % 512)

    message += "0" \* padding\_length

    print(f"Padding Length : {padding\_length + 1}")

    # Step 3: Divide the input into 512-bit blocks

    message\_words = [int(message[i: i + 32], 2) for i in range(0, len(message), 32)]

    original\_message\_length = len(message) - 64

    message\_words.append(original\_message\_length)

    # Step 4: Initialise the chaining variables

    # Hexadecimal Constants are initialized

    K = [0x67452301, 0xEFCDAB89, 0x98BADCFE, 0x10325476]

    print(f"Chaining variables are: {K}")

    def F(x, y, z):

        return (x & y) | (~x & z)

    # Step 5: Process Block

    a, b, c, d = K

    for i in range(0, len(message\_words), 16):

        f = F(b, c, d)

        g = (i >> 2) & 0x03

        for j in range(16):

            if i + j < len(message\_words):

                temp = (a + f + message\_words[i + j] + g) & 0xFFFFFFFF

                a = d

                d = c

                c = b

                b = (b + left\_rotate(temp, 7)) & 0xFFFFFFFF

    digest = format(a, "08x")

    digest += format(b, "08x")

    digest += format(c, "08x")

    digest += format(d, "08x")

    return digest

random\_message = "".join([random.choice(["0", "1"]) for \_ in range(1000)])

print("Random 1000-bit message:", random\_message)

first\_round\_res = md5(random\_message)

print("After First Round :", first\_round\_res)

**OUTPUT**

