Name: Sadhana Smruthi S   Register Number: 312217104134   Date: 18-12-2020

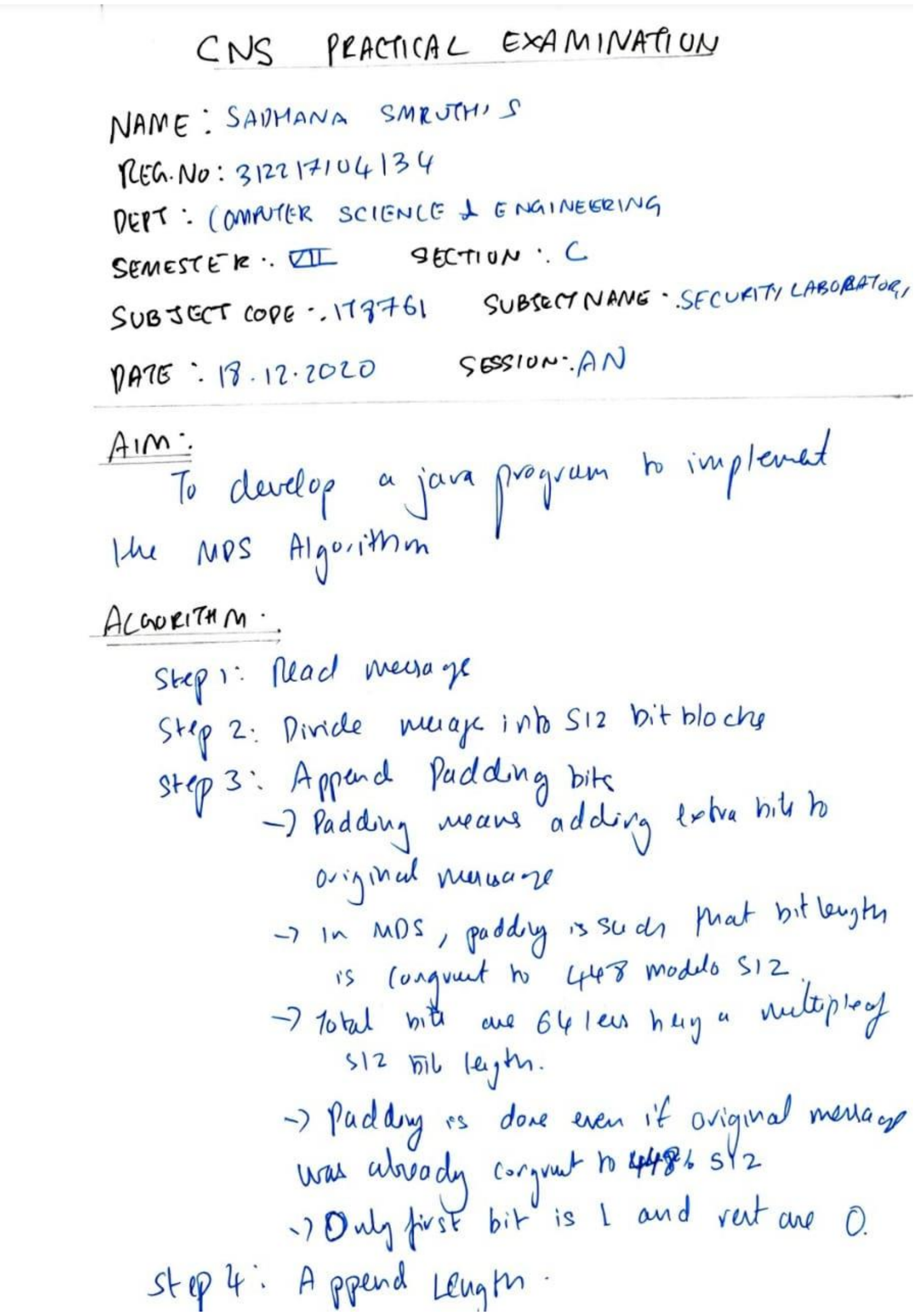
IT 8761: NETWORK SECURITY LABORATORY

AIM:

To develop a java program to implement the MD5 Algorithm

ALGORITHM:

In Answer Paper.



CNS PRACTICAL EXAMINATION

NAME : SADHANA SMRUTHI S
REG.NO : 312217104134
DEPT : COMPUTER SCIENCE & ENGINEERING
SEMESTER : VII   SECTION : C
SUBJECT CODE : IT8761   SUBJECT NAME : SECURITY LABORATORY
DATE : 18.12.2020   SESSION : AN

AIM:

To develop a java program to implement the MD5 Algorithm

ALGORITHM:

Step 1: Read message
Step 2: Divide message into 512 bit blocks
Step 3: Append Padding bits
-> Padding means adding extra bits to original message
-> In MD5, padding is such that bit length is congruent to 448 modulo 512.
-> Total bits are 64 less being a multiple of 512 bit length.
-> Padding is done even if original message was already congruent to 448 % 512
-> Only first bit is 1 and rest are 0.
Step 4: Append Length.

-> After padding add 64 bits in the end to record length of original input

-> Resultant message has length multiple of 512 bits.

Step 5: Initialize MD Buffer

-> A four word buffer (A,B,C,D) is used to compute values for message digest.

-> A,B,C & D are 32 bit registers.

Step 6: $F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$

$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$

$H(B,C,D) = B \oplus C \oplus D$

$I(B,C,D) = C \oplus (B \vee \neg D)$

Use the above compress functions in each stage.

Step 7: Display the message digest from the buffers.

METHODS & PACKAGES USED:

String makeMdDigest(input)

1. String ~~encrypt~~ makeMdDigest (String input) :

-> Takes input string of any length

-> Generates a 32 length hexadecimal string that is the 512 bit message digest.

2. public static void main (String[] args) :

-> Main function in class Main

-> Accepts string input from user

-> Prints Message digest as output .

PACKAGES & ASSOCIATED FUNCTIONS :

1. BigInteger :

-> java.math.BigInteger is a math package in java

-> Used to create number from Digest number, a 512 bit number

-> As the digest is huge, we used a special BigInteger package to store the value .

2. MessageDigest :

-> java.security.MessageDigest is a java security package

-> Used to create a message digest instance of MD5 using getInstance() method with Argument "MD5"

-> used to convert inputbytes into byte[] digest using MessageDigest.digest() function .

3. NoSuchAlgorithm :

-> java.security.NoSuchAlgorithmException

-> Is an exception passage that throws an exception when an incorrect algorithm is fed into messagedigest

i.e. MessageDigest.getInstance ("Incorrect");

will throw an error

only accepted algorithm names like MD5 or SHA-1 are allowed.

4. Scanner:

-> java.util.Scanner is a util package

-> used to get input from user using.

Scanner input = new Scanner (System.in)

input.~~get~~t line() or input.next()

generates inputs.

SAMPLE INPUT & OUTPUT:

Enter Input string:

Sadhana

Message Digest: < 32 length hexadecimal string = 128 bits >

Output received:

Message Digest: f80afe97972a38f5d431 393409b6503

RESULT:

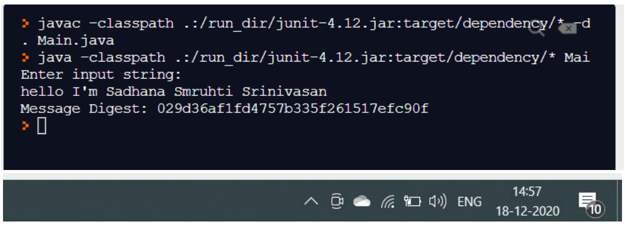
MD5 algorithm was studied & successfully executed.

SOURCE CODE:

```java
//importing all needed packages
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Scanner;

class Main {
  public static String makeMdDigest(String input){
    try{
      BigInteger numberFromDigest;
      //create messagedigest instance
      MessageDigest m=MessageDigest.getInstance("MD5");

      //make the input into bytes and convert into md
      byte[] digest=m.digest(input.getBytes());

      //make the bytes digest into signum
      numberFromDigest=new BigInteger(1,digest);

      //make the num into hex
      String hexText=numberFromDigest.toString(16);//16 means hexa
      while(hexText.length()<32){
        hexText="0"+hexText;
      }
      return hexText;
    }
    catch(NoSuchAlgorithmException e){
      throw new RuntimeException(e);
    }
  }

  public static void main(String[] args) {

    Scanner input=new Scanner(System.in);
    System.out.println("Enter input string:");
    String inputMessage=input.nextLine();
    System.out.println("Message Digest: "+makeMdDigest(inputMessage));

  }
}
```

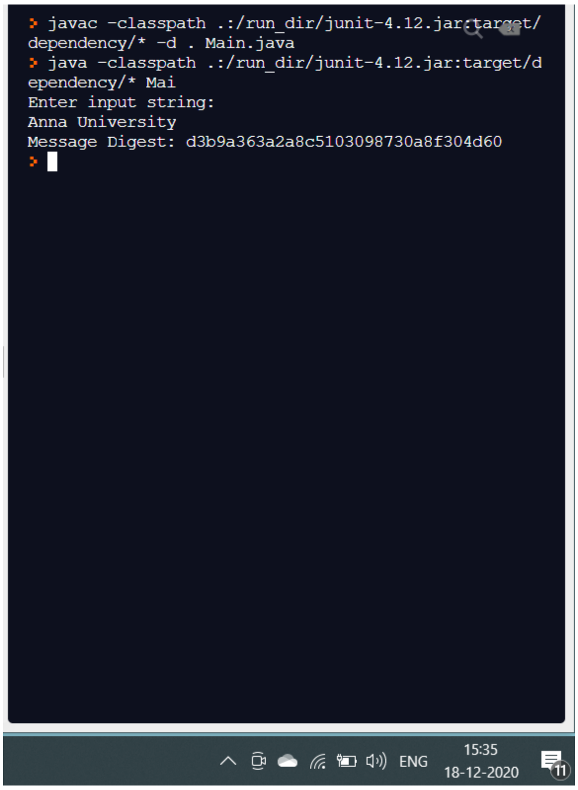
OUTPUTS:

Output 1:



```
> javac -classpath .:/run_dir/junit-4.12.jar:target/dependency/* -d
. Main.java
> java -classpath .:/run_dir/junit-4.12.jar:target/dependency/* Mai
Enter input string:
hello I'm Sadhana Smruhti Srinivasan
Message Digest: 029d36af1fd4757b335f261517efc90f
> 
```

Output 2:



```
> javac -classpath .:/run_dir/junit-4.12.jar:target/
dependency/* -d . Main.java
> java -classpath .:/run_dir/junit-4.12.jar:target/d
ependency/* Mai
Enter input string:
Anna University
Message Digest: d3b9a363a2a8c5103098730a8f304d60
> 
```

RESULT:

The MD5 algorithm was successfully executed using java. A string was encrypted to produce a message digest that is 128 bits long (32 HexaDecimal digits).