Certificate

THIS IS CERTIFY TO THE WORK EMBODIES IN THE "**LAB PRACTICE-2**" PRACTICAL.

THIS ARE BONAFIDE STUDENTS OF THIS INSTITUTE AND THE WORK HAS BEEN CARRIED OUT BY THEM UNDER THE GUIDANCE OF "**Prof.PANSARE P.P.** " AND IT IS APPROVED FOR THE PARTIAL FULL-FILLMENT OF THE REQUIREMENT OF SAVITRIBAI PHULE PUNE UNIVERSITY FOR THE DEGREE OF BACHELOR OF ENGINEERING SECOND YEAR OF COMPUTER ENGINEERING.

DATE:- **PLACE**:- Belhe

NAME:-NAVALE RUPESH PANDHARINATH

ROLLNO:-36 **BATCH**:-B

Prof. Pansare P. P.
(Dept. of Computer Engineering)

Prof. Shegar S.R.
(Head of Dept. of Computer Engineering)

Dr. Narawade N.S.
(Principle of SGOI COE)

INFORMATION SECURITY

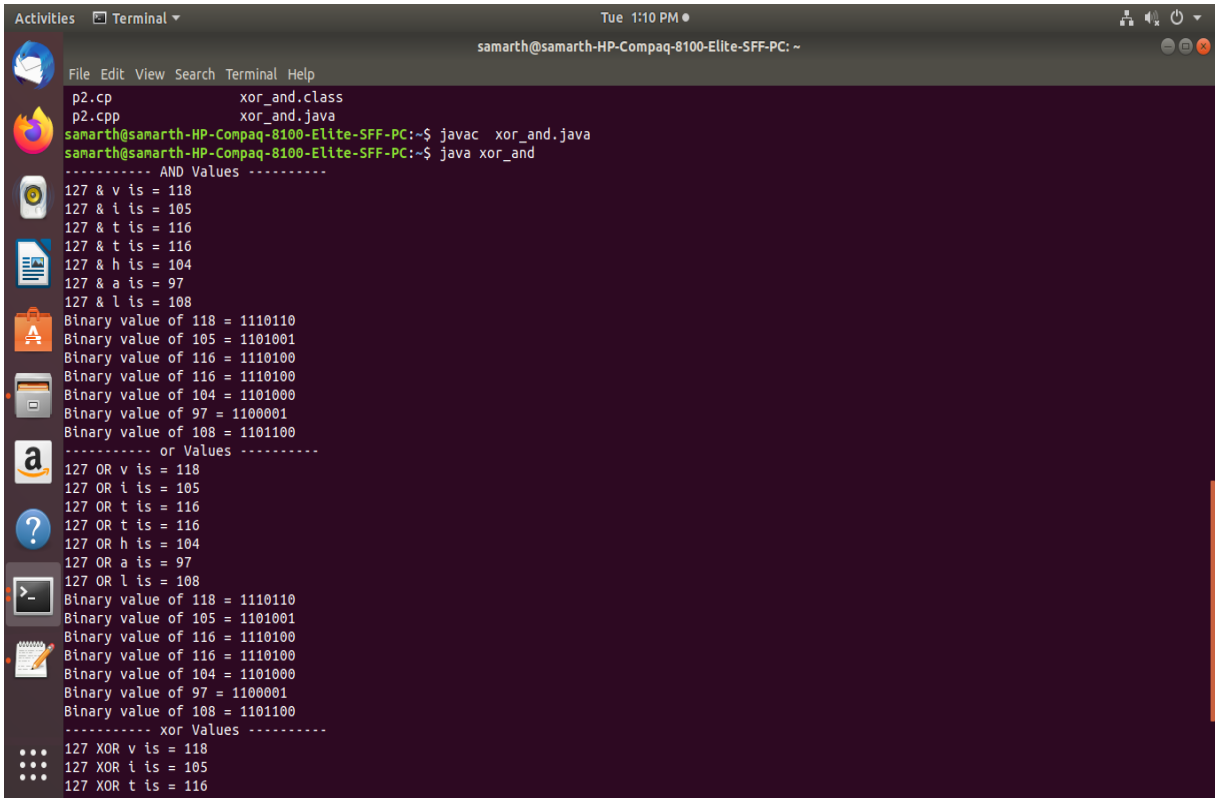
EXPERIMENT:01

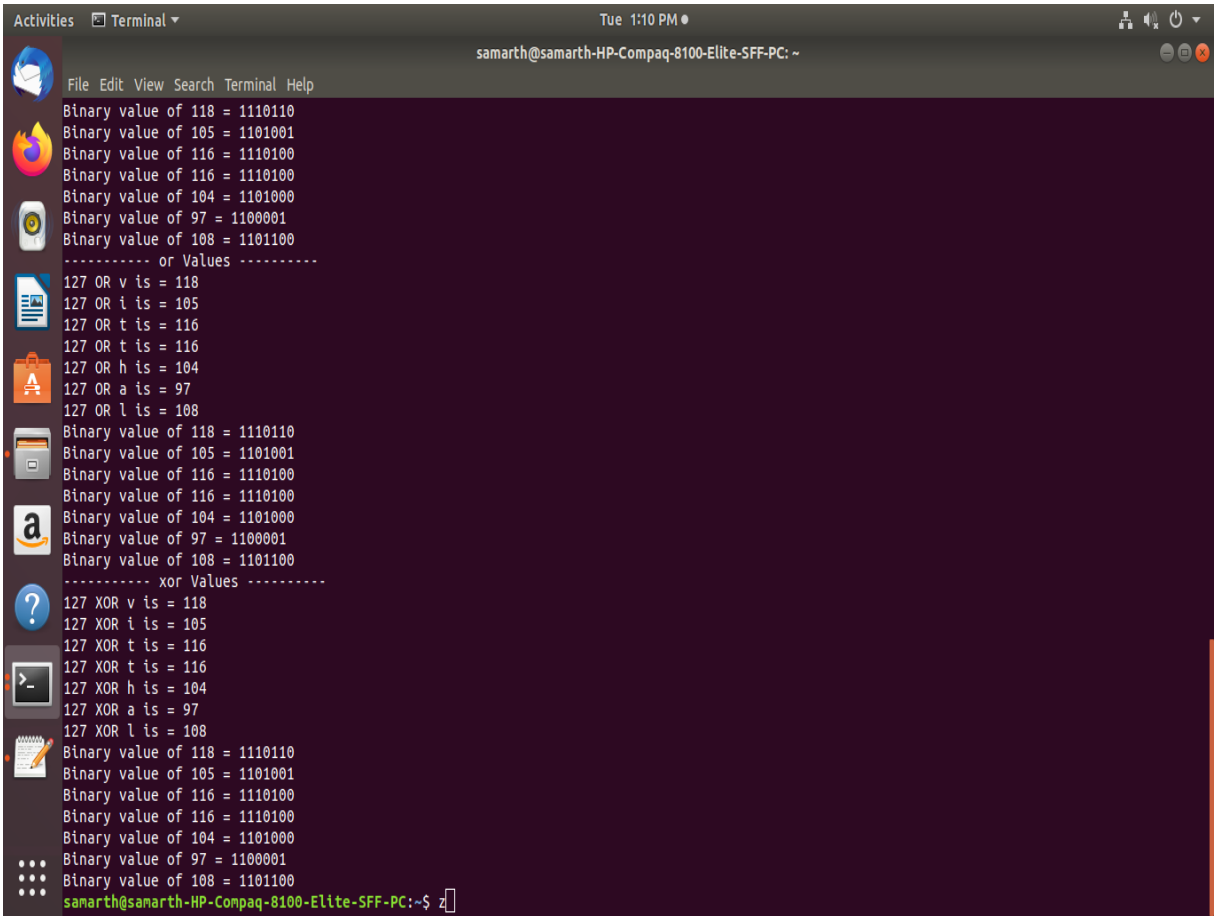
AND XOR Program /*Write a Java/C/C++/Python program that contains a string (char pointer) with a value \Hello World'. The program should AND or and XOR each character in this string with 127 and display the result.*/

Program:

```
public class xor_and {
public static void main(String s[]) {
String Message = "vitthal";
int var1 = 127;
int var2[] = new int[20];
//and
  System.out.println("----------- AND Values ----------");
 for (int i = 0; i < Message.length(); i++) {
  var2[i] = 127 & Message.charAt(i);
    System.out.println("127 & " + Message.charAt(i) + " is = " + var2[i]);
 }
   for (int i = 0; i < Message.length(); i++) {
   System.out.println("Binary value of " + var2[i] + " = " +
Integer.toBinaryString(var2[i]));
 }
//or
 System.out.println("----------- or Values ----------");
 for (int i = 0; i < Message.length(); i++) {
  var2[i] = 127 & Message.charAt(i);
    System.out.println("127 OR " + Message.charAt(i) + " is = " + var2[i]);
 }
   for (int i = 0; i < Message.length(); i++) {
   System.out.println("Binary value of " + var2[i] + " = " +
Integer.toBinaryString(var2[i]));
 }
//XOR
 System.out.println("----------- xor Values ----------");
 for (int i = 0; i < Message.length(); i++) {
  var2[i] = 127 & Message.charAt(i);
    System.out.println("127 XOR " + Message.charAt(i) + " is = " + var2[i]);
 }
   for (int i = 0; i < Message.length(); i++) {
   System.out.println("Binary value of " + var2[i] + " = " +
Integer.toBinaryString(var2[i]));
 }
}
}
```

Output:


Activities
Terminal
Tue 1:10 PM
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC: ~
File Edit View Search Terminal Help
p2.cp xor_and.class
p2.cpp xor_and.java
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC:~$ javac xor_and.java
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC:~$ java xor_and
---------- AND Values ----------
127 & v is = 118
127 & i is = 105
127 & t is = 116
127 & t is = 116
127 & h is = 104
127 & a is = 97
127 & l is = 108
Binary value of 118 = 1110110
Binary value of 105 = 1101001
Binary value of 116 = 1110100
Binary value of 116 = 1110100
Binary value of 104 = 1101000
Binary value of 97 = 1100001
Binary value of 108 = 1101100
---------- or Values ----------
127 OR v is = 118
127 OR i is = 105
127 OR t is = 116
127 OR t is = 116
127 OR h is = 104
127 OR a is = 97
127 OR l is = 108
Binary value of 118 = 1110110
Binary value of 105 = 1101001
Binary value of 116 = 1110100
Binary value of 116 = 1110100
Binary value of 104 = 1101000
Binary value of 97 = 1100001
Binary value of 108 = 1101100
---------- xor Values ----------
127 XOR v is = 118
127 XOR i is = 105
127 XOR t is = 116


Activities
Terminal
Tue 1:10 PM
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC: ~
File Edit View Search Terminal Help
Binary value of 118 = 1110110
Binary value of 105 = 1101001
Binary value of 116 = 1110100
Binary value of 116 = 1110100
Binary value of 104 = 1101000
Binary value of 97 = 1100001
Binary value of 108 = 1101100
---------- or Values ----------
127 OR v is = 118
127 OR i is = 105
127 OR t is = 116
127 OR t is = 116
127 OR h is = 104
127 OR a is = 97
127 OR l is = 108
Binary value of 118 = 1110110
Binary value of 105 = 1101001
Binary value of 116 = 1110100
Binary value of 116 = 1110100
Binary value of 104 = 1101000
Binary value of 97 = 1100001
Binary value of 108 = 1101100
---------- xor Values ----------
127 XOR v is = 118
127 XOR i is = 105
127 XOR t is = 116
127 XOR t is = 116
127 XOR h is = 104
127 XOR a is = 97
127 XOR l is = 108
Binary value of 118 = 1110110
Binary value of 105 = 1101001
Binary value of 116 = 1110100
Binary value of 116 = 1110100
Binary value of 104 = 1101000
Binary value of 97 = 1100001
Binary value of 108 = 1101100
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC:~$ z

EXPERIMENT NO:-02

Write a Java/C/C++/Python program to perform encryption and decryption using the method of Transposition technique.

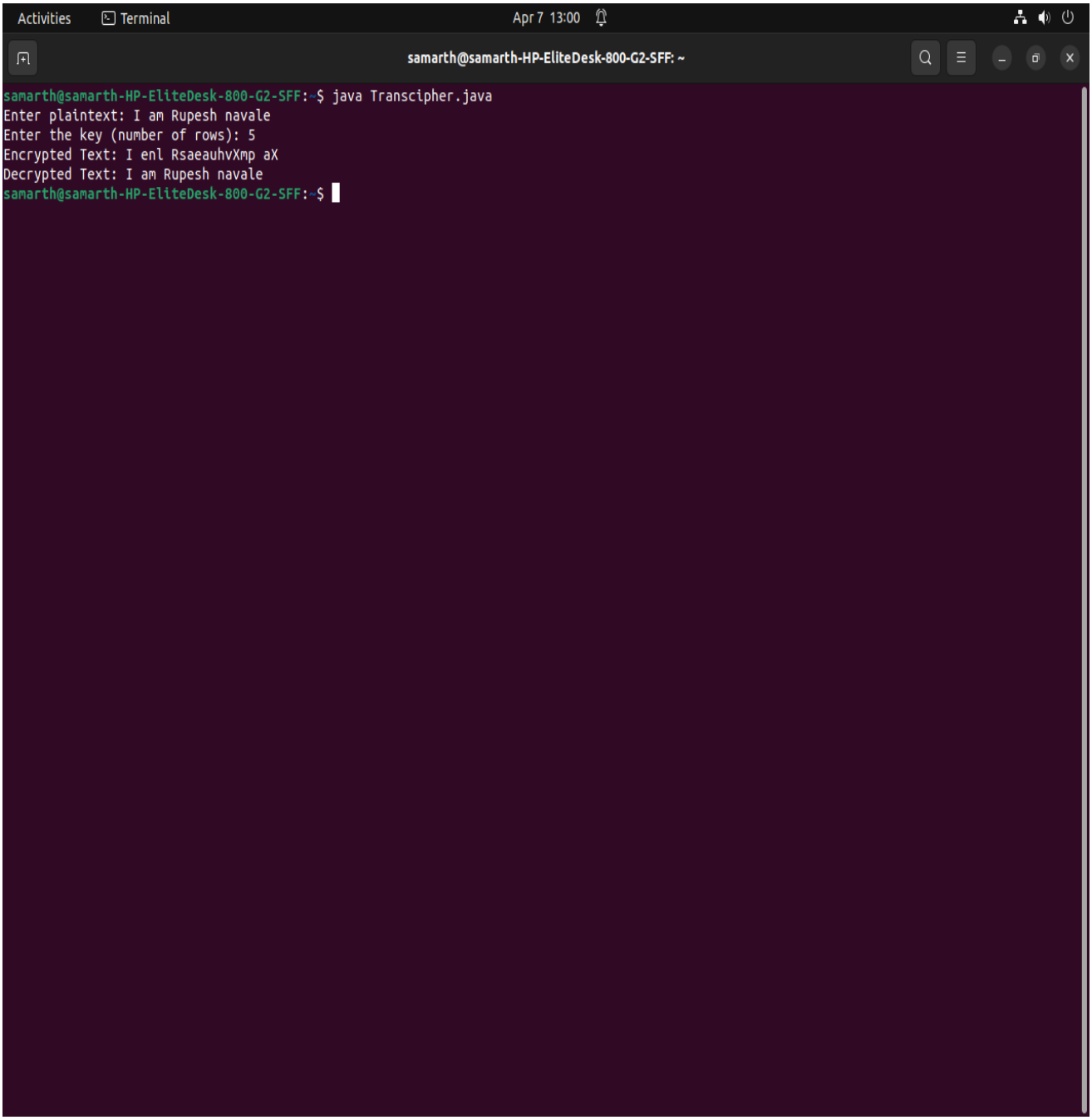
Program:-

```
import java.util.Scanner;
public class trans {
public static String encrypt(String text, int key) {
char[][] grid = new char[key][(int) Math.ceil((double) text.length() / key)];
int index = 0;
for (int i = 0; i < key; i++) {
for (int j = 0; j < grid[i].length; j++) {
if (index < text.length()) {
grid[i][j] = text.charAt(index++);
} else {
grid[i][j] = 'X';
}
}
}
StringBuilder ciphertext = new StringBuilder();
for (int j = 0; j < grid[0].length; j++) {
for (int i = 0; i < key; i++) {
ciphertext.append(grid[i][j]);
}
}
return ciphertext.toString();
}
public static String decrypt(String ciphertext, int key) {
int numRows = key;
```

```
int numCols = (int) Math.ceil((double) ciphertext.length() / numRows);
char[][] grid = new char[numRows][numCols];
int index = 0;
for (int j = 0; j < numCols; j++) {
for (int i = 0; i < numRows; i++) {
if (index < ciphertext.length()) {
grid[i][j] = ciphertext.charAt(index++);
}
}
}
StringBuilder plaintext = new StringBuilder();
for (int i = 0; i < numRows; i++) {
for (int j = 0; j < numCols; j++) {
if (grid[i][j] != 'X') {
plaintext.append(grid[i][j]);
}
}
}
return plaintext.toString();
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter plaintext: ");
String plaintext = scanner.nextLine();
System.out.print("Enter the key (number of rows): ");
int key = scanner.nextInt();
String ciphertext = encrypt(plaintext, key);
System.out.println("Encrypted Text: " + ciphertext);
String decryptedText = decrypt(ciphertext, key);
```

```
System.out.println("Decrypted Text: " + decryptedText);
scanner.close();
}
}
```

Output:-



EXPERIMENT NO:-03

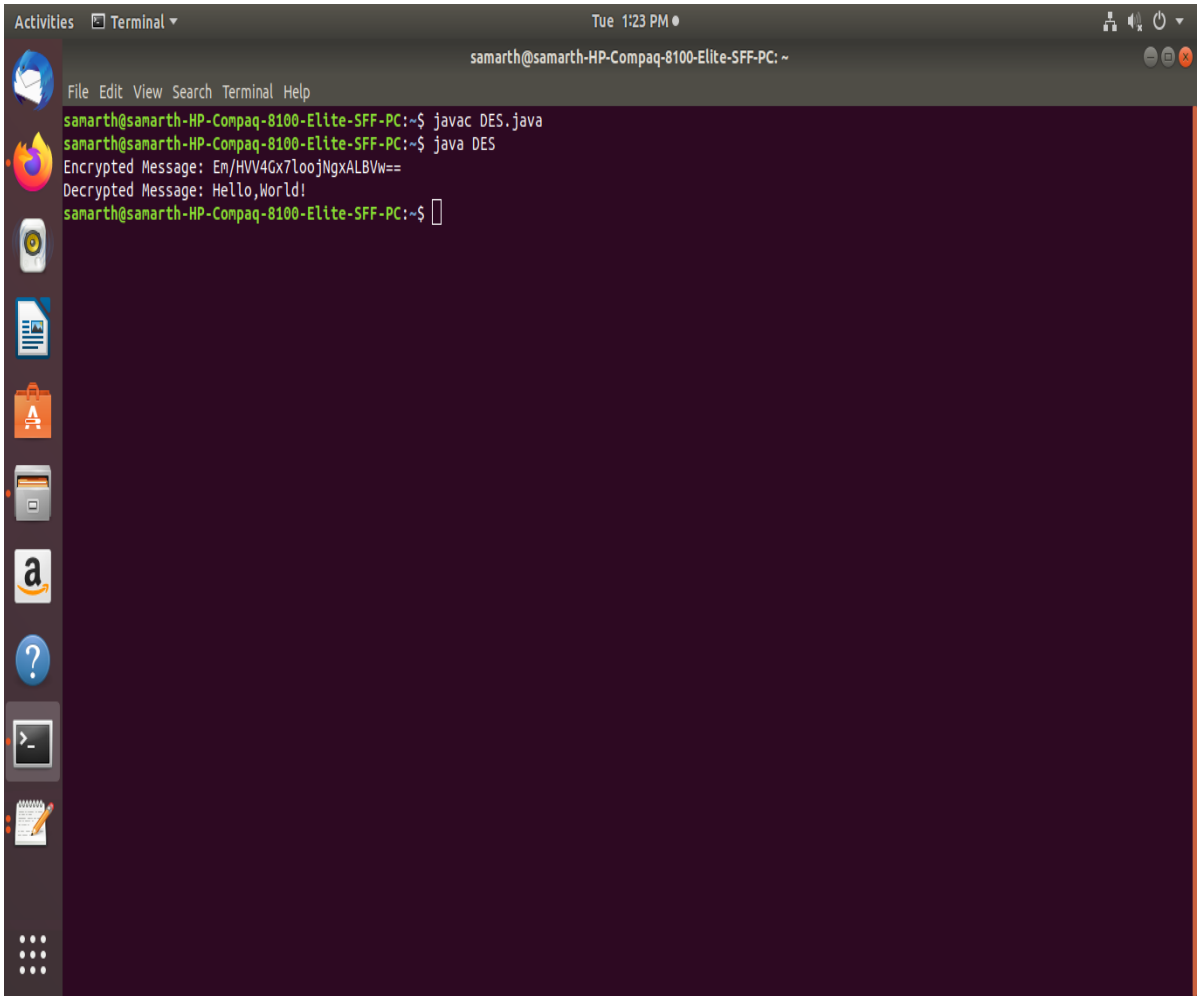
Write a Java/C/C++/Python program to implement DES algorithm.

Program:

```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;
public class DES{
public static String encrypt(String message,String key) throws Exception {
SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(),"DES");
Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE,secretKey);
byte[] encryptedBytes = cipher.doFinal(message.getBytes());
return Base64.getEncoder().encodeToString(encryptedBytes);
}
public static String decrypt(String encryptedMessage, String key) throws
Exception {
SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(),"DES");
Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
cipher.init(Cipher.DECRYPT_MODE, secretKey);
byte[] decodedBytes = Base64.getDecoder().decode(encryptedMessage);
byte[] decryptedBytes = cipher.doFinal(decodedBytes);
return new String(decryptedBytes);
}
public static void main(String[] args) {
try {
String message = "Hello,World!";
String key = "12345678";
```

```
String encryptedMessage = encrypt(message, key);
System.out.println("Encrypted Message: " + encryptedMessage);
String decryptedMessage = decrypt(encryptedMessage, key);
System.out.println("Decrypted Message: " + decryptedMessage);
} catch (Exception e)
{
e.printStackTrace();
}
}
}
```

Output:-



EXPERIMENT NO:-04

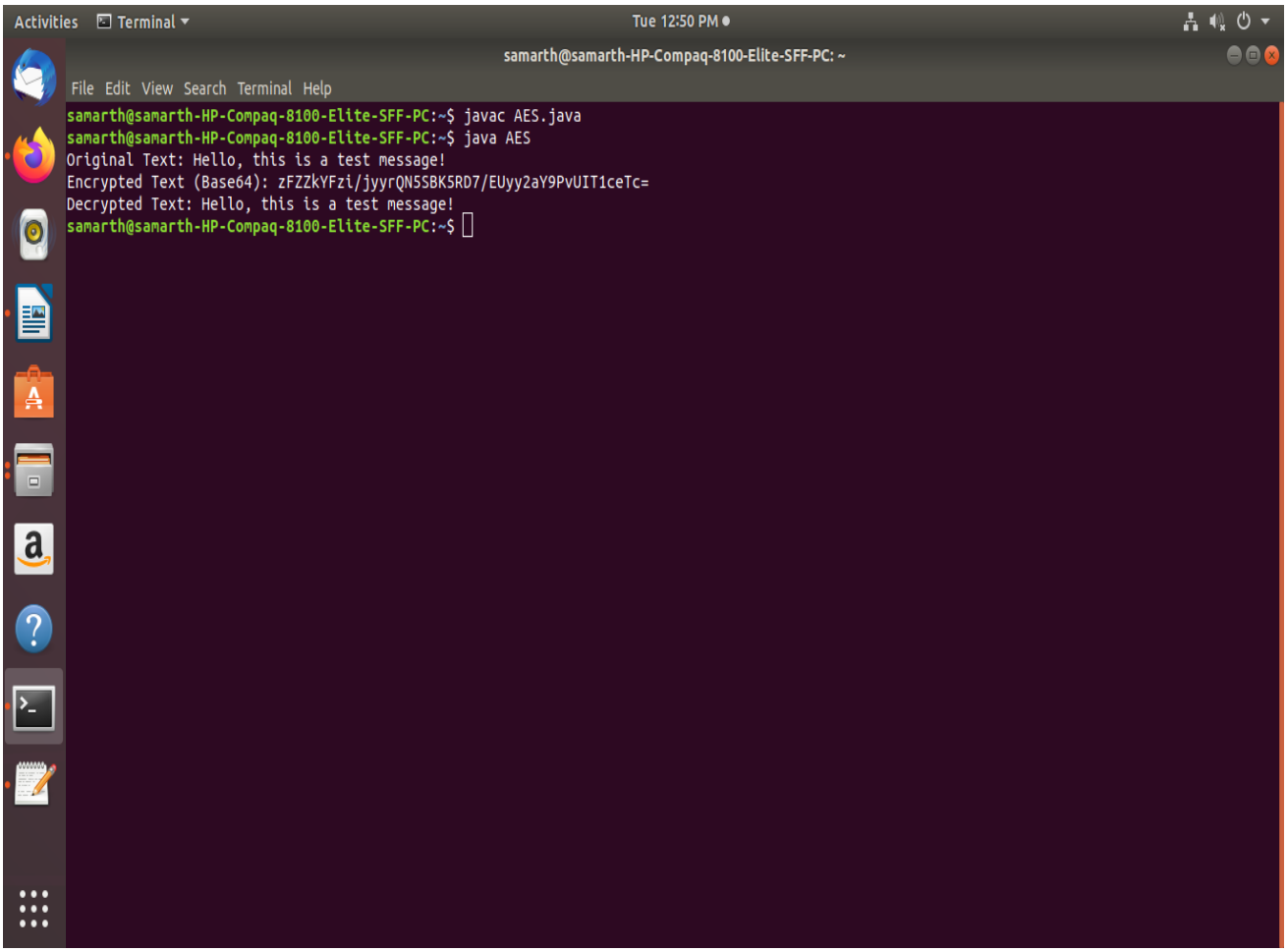
Write a Java/C/C++/Python program to implement AES algorithm.

Program:-

```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;
public class AES {
public static String encrypt(String plainText, String secretKey) throws
Exception {
SecretKeySpec key = new SecretKeySpec(secretKey.getBytes(), "AES");
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.ENCRYPT_MODE, key);
byte[] encryptedBytes = cipher.doFinal(plainText.getBytes());
return Base64.getEncoder().encodeToString(encryptedBytes);
}
public static String decrypt(String encryptedText, String secretKey) throws
Exception {
SecretKeySpec key = new SecretKeySpec(secretKey.getBytes(), "AES");
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.DECRYPT_MODE, key);
byte[] decodedBytes = Base64.getDecoder().decode(encryptedText);
byte[] decryptedBytes = cipher.doFinal(decodedBytes);
return new String(decryptedBytes);
}
public static void main(String[] args) {
try {
String secretKey = "1234567890123456";
```

```
String plainText = "Hello, this is a test message!";
System.out.println("Original Text: " + plainText);
String encryptedText = encrypt(plainText, secretKey);
System.out.println("Encrypted Text (Base64): " + encryptedText);
String decryptedText = decrypt(encryptedText, secretKey);
System.out.println("Decrypted Text: " + decryptedText);
} catch (Exception e) {
e.printStackTrace();
}
}
}
```

Output:-



EXPERIMENT NO:-05

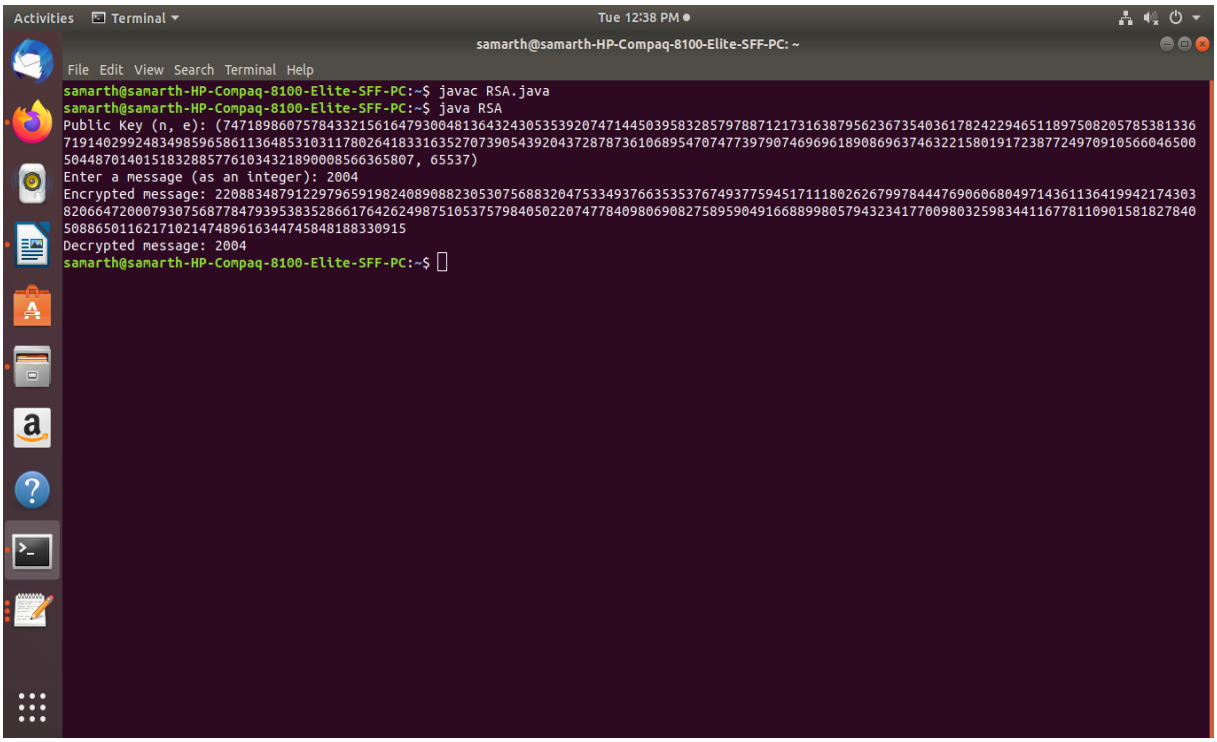
Write a Java/C/C++/Python program to implement RSA algorithm.

Program:

```
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class RSA {
    private BigInteger n, d, e;
    private int bitLength = 1024;
    private SecureRandom random = new SecureRandom();
    public RSA() {
        BigInteger p = BigInteger.probablePrime(bitLength / 2, random);
        BigInteger q = BigInteger.probablePrime(bitLength / 2, random);
        n = p.multiply(q);
        BigInteger phi =
(p.subtract(BigInteger.ONE)).multiply(q.subtract(BigInteger.ONE));

     e = new BigInteger("65537")
        d = e.modInverse(phi);
    }
    public BigInteger encrypt(BigInteger message) {
        return message.modPow(e, n);
    }
    public BigInteger decrypt(BigInteger ciphertext) {
        return ciphertext.modPow(d, n);
    }
    public BigInteger getN() {
        return n;
```

```
    }
    public BigInteger getE() {
        return e;
    }
    public static void main(String[] args) {
        RSA rsa = new RSA();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Public Key (n, e): (" + rsa.getN() + ", " + rsa.getE() +
")");
        System.out.print("Enter a message (as an integer): ");
        BigInteger message = scanner.nextBigInteger();
        BigInteger encrypted = rsa.encrypt(message);
        System.out.println("Encrypted message: " + encrypted);
        BigInteger decrypted = rsa.decrypt(encrypted);
        System.out.println("Decrypted message: " + decrypted);
        scanner.close();
    }
```

Output:



Artificial Inteligence

EXPERIMENT NO:-06

Implement DFS & BFS algorithm & develop recursive algorithm for a searching all vertices of graph.

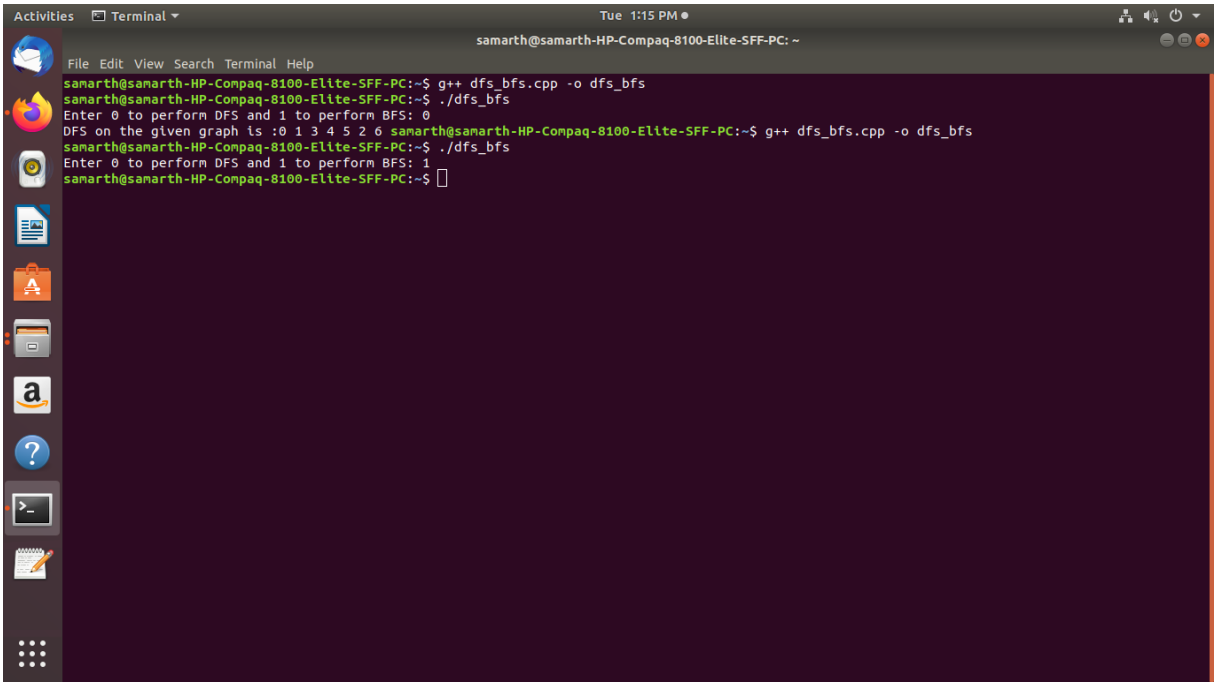
Program:

```
#include<iostream>
#include<list>
#include<map>
#include<queue>
using namespace std;
class Graph{
public:
map<int, list<int>> adjList;
map<int, bool> visited;
queue<int> q;
//Copy Constructor
// Graph(const Graph &g){
// }
void addEdge(int src, int dest){
adjList[src].push_back(dest);
adjList[dest].push_back(src);
}
void DFS(int node){
//Mark Node as visited
visited[node] = true;
//Print Node
cout << node << " ";
//Vist its neighbours and recurse
```

```
for(int i : adjList[node]){
//If node is not visited
if(!visited[i]) DFS(i);
}
}
void BFS(){

if(q.empty()) return;

int node = q.front();
q.pop();

cout << node << " ";
for(int i : adjList[node]){
if(!visited[i]){
visited[i] = true;
q.push(i);
}
}
BFS();
}
};
int main(){
Graph g;
g.addEdge(0,1);
g.addEdge(0,2);
g.addEdge(0,3);
g.addEdge(1,3);
```

```
g.addEdge(3,4);
g.addEdge(4,5);
g.addEdge(2,6);
int ch;
cout << "Enter 0 to perform DFS and 1 to perform BFS: ";
cin >> ch;if(!ch){
cout << "DFS on the given graph is :";
g.DFS(0);
}
else{
cout << "BFS on the given graph is: ";
g.q.push(0);
g.visited[0] = true;
g.BFS();
}
return 0;
}
```

Output:



EXPERIMENT NO:-07

Implement A star Algorithm for any game search problem.

Program:

```
#include<iostream>
#include<cmath>
#include<limits.h>
using namespace std;
int g = 0;
void Print(int puzzle[]) {
    for(int i = 0; i < 9; i++) {
        if(i % 3 == 0) cout << '\n';
        if(puzzle[i] == -1) cout << "_ ";
        else cout << puzzle[i] << " ";
    }
    cout << "\n\n";
}
void moveLeft(int start[], int position) {
    swap(start[position], start[position - 1]);
}
void moveRight(int start[], int position) {
    swap(start[position], start[position + 1]);
}
void moveUp(int start[], int position) {
    swap(start[position], start[position - 3]);
}
void moveDown(int start[], int position) {
```

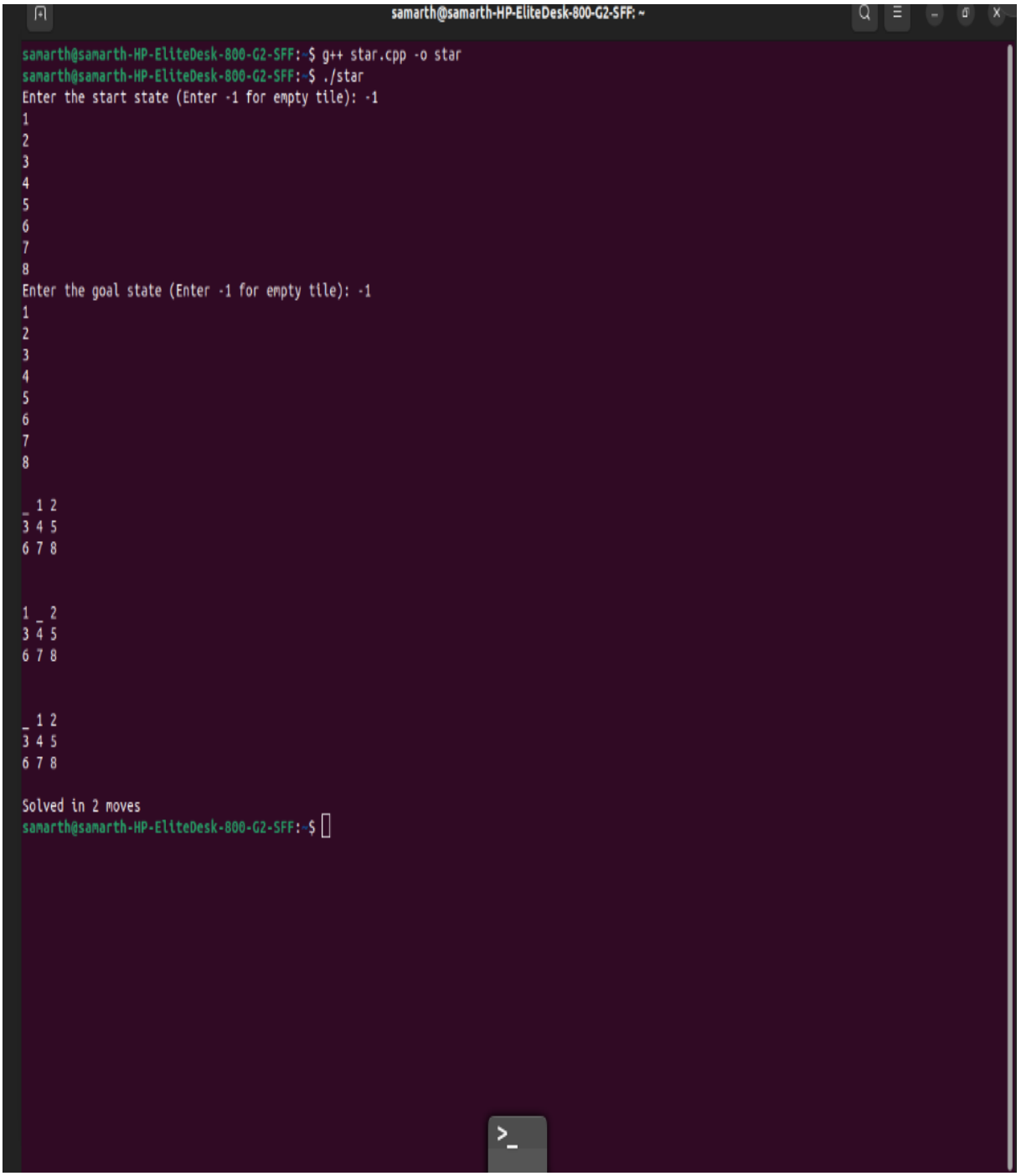
```
    swap(start[position], start[position + 3]);
}
void Copy(int temp[], int real[]) {
    for(int i = 0; i < 9; i++) temp[i] = real[i];
}
int heuristic(int start[], int goal[]) {
    int h = 0;
    for(int i = 0; i < 9; i++) {
        for(int j = 0; j < 9; j++) {
            if (start[i] == goal[j] && start[i] != -1) {
                h += abs((j - i) / 3) + abs((j - i) % 3);
            }
        }
    }
    return h + g;
}
void moveTile(int start[], int goal[]) {
    int emptyAt = 0;
    for(int i = 0; i < 9; i++) {
        if(start[i] == -1) {
            emptyAt = i;
            break;
        }
    }
 int t1[9], t2[9], t3[9], t4[9];
    int f1 = INT_MAX, f2 = INT_MAX, f3 = INT_MAX, f4 = INT_MAX;
    Copy(t1, start);
    Copy(t2, start);
```

```
    Copy(t3, start);
    Copy(t4, start);
    int row = emptyAt / 3;
    int col = emptyAt % 3;
 if(col - 1 >= 0) {
        moveLeft(t1, emptyAt);
        f1 = heuristic(t1, goal);
    }
if(col + 1 < 3) {
        moveRight(t2, emptyAt);
        f2 = heuristic(t2, goal);
    }
if(row + 1 < 3) {
        moveDown(t3, emptyAt);
        f3 = heuristic(t3, goal);
    }
if(row - 1 >= 0) {
        moveUp(t4, emptyAt);
        f4 = heuristic(t4, goal);
    }
 if(f1 <= f2 && f1 <= f3 && f1 <= f4) {
        moveLeft(start, emptyAt);
    }
    else if(f2 <= f1 && f2 <= f3 && f2 <= f4) {
        moveRight(start, emptyAt);
    }
    else if(f3 <= f1 && f3 <= f2 && f3 <= f4) {
        moveDown(start, emptyAt);
```

```
    }
    else {
        moveUp(start, emptyAt);
    }
}

void solveEight(int start[], int goal[]) {
    g++;
    moveTile(start, goal);
    Print(start);
    int f = heuristic(start, goal);
    if(f == g) {
        cout << "Solved in " << f << " moves\n";
        return;
    }
    solveEight(start, goal);
}
bool solvable(int start[]) {
    int invrs = 0;
    for(int i = 0; i < 9; i++) {
        if(start[i] <= 1) continue;
        for(int j = i + 1; j < 9; j++) {
            if(start[j] == -1) continue;
            if(start[i] > start[j]) invrs++;
        }
    }
    return invrs % 2 == 0;
}
```

```
int main() {
    int start[9];
    int goal[9];
    cout << "Enter the start state (Enter -1 for empty tile): ";
    for(int i = 0; i < 9; i++) {
        cin >> start[i];
    }

    cout << "Enter the goal state (Enter -1 for empty tile): ";
    for(int i = 0; i < 9; i++) {
        cin >> goal[i];
    }
    Print(start);
  if(solvable(start)) {
        solveEight(start, goal);
    }
    else {
        cout << "\nImpossible To Solve\n";
    }
    return 0;
}
```

Output:


samarth@samarth-HP-EliteDesk-800-G2-SFF: ~
samarth@samarth-HP-EliteDesk-800-G2-SFF:~$ g++ star.cpp -o star
samarth@samarth-HP-EliteDesk-800-G2-SFF:~$ ./star
Enter the start state (Enter -1 for empty tile): -1
1
2
3
4
5
6
7
8
Enter the goal state (Enter -1 for empty tile): -1
1
2
3
4
5
6
7
8
_ 1 2
3 4 5
6 7 8
1 _ 2
3 4 5
6 7 8
_ 1 2
3 4 5
6 7 8
Solved in 2 moves
samarth@samarth-HP-EliteDesk-800-G2-SFF:~$

EXPERIMENT NO:-08

Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.

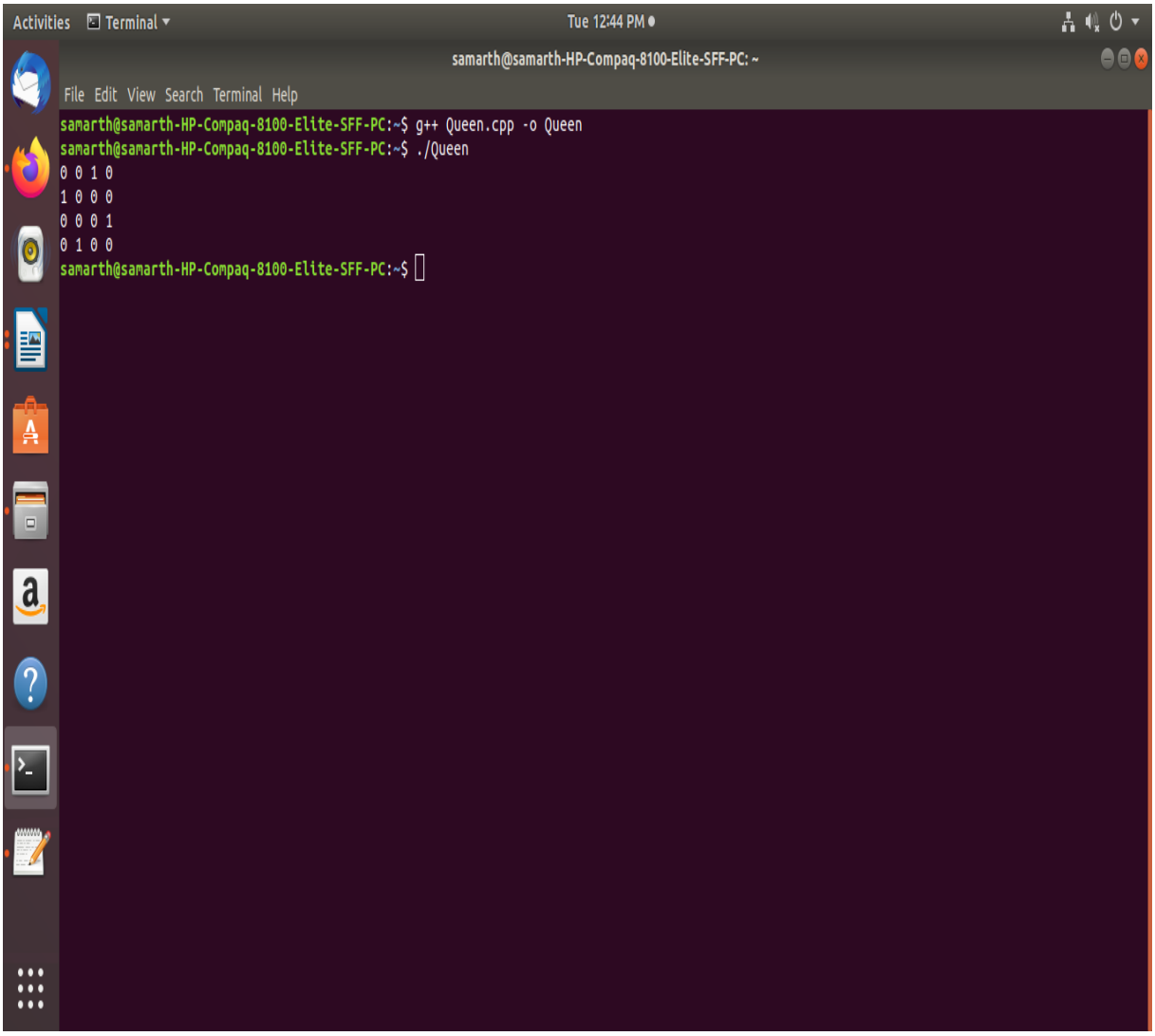
Program:

```
#include <iostream>
#include <cstring>

#define N 4

using namespace std;

void printPuzzle(int board[N][N]) {
   for (int i = 0; i < N; i++) {
      for (int j = 0; j < N; j++) {
         cout << board[i][j] << " ";
      }
      cout << '\n';
   }
}

bool isSafe(int row, int col, int slash[N][N], int backSlash[N][N],
         bool rowLookup[], bool slashLookUp[], bool backSlashLookUp[]) {

   if (slashLookUp[slash[row][col]] ||
      backSlashLookUp[backSlash[row][col]] ||
      rowLookup[row]) {
      return false;
   }

   return true;
}

bool solveNqueenUtil(int board[N][N], int col, int slash[N][N], int
backSlash[N][N],
                  bool rowLookUp[N], bool slashLookUp[N], bool
backSlashLookUp[N]) {

   if (col >= N) return true;
   for (int i = 0; i < N; i++) {
```

```
        if (isSafe(i, col, slash, backSlash, rowLookUp, slashLookUp,
backSlashLookUp)) {


            board[i][col] = 1;
            rowLookUp[i] = true;
            slashLookUp[slash[i][col]] = true;
            backSlashLookUp[backSlash[i][col]] = true;

            if (solveNqueenUtil(board, col + 1, slash, backSlash, rowLookUp,
slashLookUp, backSlashLookUp))
                return true;


            board[i][col] = 0;
            rowLookUp[i] = false;
            slashLookUp[slash[i][col]] = false;
            backSlashLookUp[backSlash[i][col]] = false;
        }
    }


    return false;
}

void solveNqueen() {
    int board[N][N];
    memset(board, 0, sizeof(board));

    int backSlash[N][N];
    int slash[N][N];


    bool rowLookUp[N] = {false};
    bool backSlashLookUp[2 * N - 1] = {false};
    bool slashLookUp[2 * N - 1] = {false};


    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            backSlash[i][j] = (i - j) + (N - 1);
            slash[i][j] = i + j;
        }
    }
```

```
    if (solveNqueenUtil(board, 0, slash, backSlash, rowLookUp, slashLookUp,
backSlashLookUp) == false) {
        cout << "Solution does not exist!!\n";
        printPuzzle(board);
    } else {
        printPuzzle(board);
    }
}

int main() {
    solveNqueen();
    return 0;
}
```

Output:



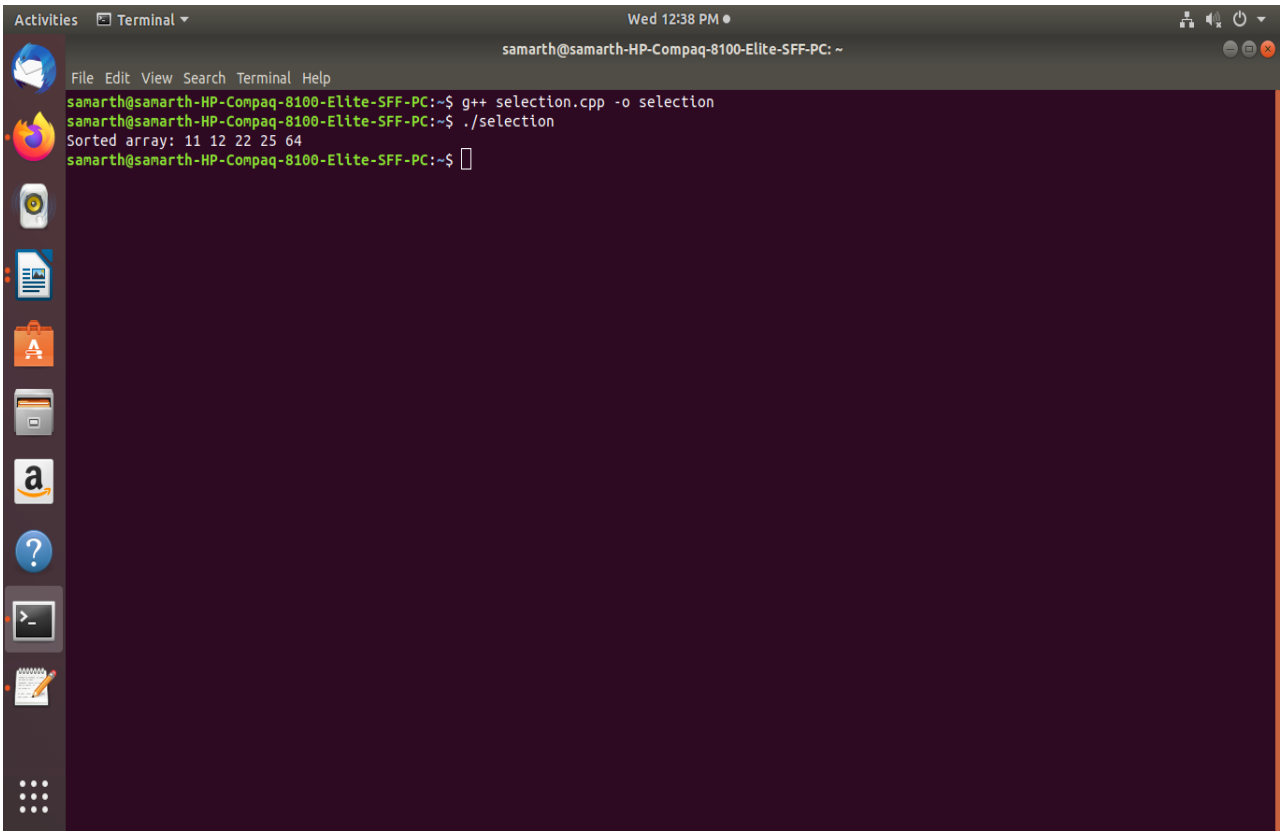
EXPERIMENT NO:-09

Implement Greedy search algorithm for any of the following application: I. Selection Sort

Program:

```
#include <iostream>
using namespace std;

void selectionSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int min_idx = i;
        for (int j = i + 1; j < n; j++) {
            if (arr[j] < arr[min_idx]) {
                min_idx = j;
            }
        }
        if (min_idx != i) {
            swap(arr[i], arr[min_idx]);
        }
    }
}

int main() {
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr) / sizeof(arr[0]);
    selectionSort(arr, n);

    cout << "Sorted array: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```

Output:


Activities
Terminal
Wed 12:38 PM
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC: ~
File Edit View Search Terminal Help
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC:~$ g++ selection.cpp -o selection
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC:~$ ./selection
Sorted array: 11 12 22 25 64
samarth@samarth-HP-Compaq-8100-Elite-SFF-PC:~$

EXPERIMENT NO:-10

Develop an elementary chatbot for any suitable customer interaction application

Program:

```
#include <iostream>
#include <string>
using namespace std;

void chatbotResponse(const string &input) {
   if (input == "hello" || input == "hi") {
      cout << "Chatbot: Hello! How can I help you today?" << endl;
   } else if (input == "help") {
      cout << "Chatbot: Sure, I'm here to assist you. What do you need help
with?" << endl;
   } else if (input == "bye" || input == "exit") {
      cout << "Chatbot: Goodbye! Have a great day!" << endl;
   } else if (input == "hours") {
      cout << "Chatbot: Our working hours are from 9 AM to 6 PM, Monday to
Friday." << endl;
   } else if (input == "contact") {
      cout << "Chatbot: You can contact us at support@example.com or call
(123) 456-7890." << endl;
   } else {
      cout << "Chatbot: I'm sorry, I didn't understand that. Could you please
rephrase?" << endl;
   }
}

int main() {
   string userInput;
   cout << "Welcome to the Customer Support Chatbot!" << endl;
   cout << "Type 'exit' to end the chat." << endl;

   while (true) {
      cout << "You: ";
      getline(cin, userInput);

      if (userInput == "exit" || userInput == "bye") {
         chatbotResponse(userInput);
         break;
      }
```

```
        chatbotResponse(userInput);
    }

    return 0;
}
```

Output: