Name: **Muhammad Ubaid**

Reg No: **4290-FBAS/BSCS/F20**

Subject: **Information Security (IS)**

Submitted to: **Sir Qaiser Javaid**

Date: **21-May, 2023**

Question NO 01: Program to add 3 in an ASCII value.Code

Answer:

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

string encryptMessage(const string& message) {
    string encryptedMessage = message;

    for (char& c : encryptedMessage) {
        if (isalpha(c)) {
            c = (c - 'a' + 3) % 26 + 'a';
        }
    }

    return encryptedMessage;
}

string decryptMessage(const string& encryptedMessage) {
    string decryptedMessage = encryptedMessage;

    for (char& c : decryptedMessage) {
        if (isalpha(c)) {
            c = (c - 'a' + 23) % 26 + 'a';
        }
    }
```

```
    return decryptedMessage;
}

int main() {
    string message;

    cout << "Enter a message: ";
    getline(cin, message);

    string encryptedMessage = encryptMessage(message);
    cout << "Encrypted message: " << encryptedMessage << endl;

    string decryptedMessage = decryptMessage(encryptedMessage);
    cout << "Decrypted message: " << decryptedMessage << endl;

    return 0;
}
```

Question NO 02: Program for Random Key Encryption.

Answer:

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

string generateRandomKey(int length) {
    string key = "";
    const string alphabet = "abcdefghijklmnopqrstuvwxyz";

    mt19937 rng(time(0));
    uniform_int_distribution<int> distribution(0, alphabet.size() - 1);

    for (int i = 0; i < length; i++) {
        int index = distribution(rng);
        key += alphabet[index];
    }

    return key;
}
```

```
string encryptMessage(const string& message, const string& key) {
    string encryptedMessage = message;

    for (int i = 0; i < encryptedMessage.length(); i++) {
        char c = encryptedMessage[i];
        if (isalpha(c)) {
            char base = islower(c) ? 'a' : 'A';
            int shift = key[i % key.length()] - 'a';
            encryptedMessage[i] = ((c - base + shift) % 26) + base;
        }
    }

    return encryptedMessage;
}

string decryptMessage(const string& encryptedMessage, const string& key) {
    string decryptedMessage = encryptedMessage;

    for (int i = 0; i < decryptedMessage.length(); i++) {
        char c = decryptedMessage[i];
        if (isalpha(c)) {
            char base = islower(c) ? 'a' : 'A';
            int shift = key[i % key.length()] - 'a';
            decryptedMessage[i] = ((c - base - shift + 26) % 26) + base;
        }
    }
```

```
    return decryptedMessage;
}

int main() {
    string message;

    cout << "Enter a message: ";
    getline(cin, message);

    string key = generateRandomKey(message.length());
    cout << "Randomly generated key: " << key << endl;

    string encryptedMessage = encryptMessage(message, key);
    cout << "Encrypted message: " << encryptedMessage << endl;

    string decryptedMessage = decryptMessage(encryptedMessage, key);
    cout << "Decrypted message: " << decryptedMessage << endl;

    return 0;
}
```

Output

```
/tmp/4nH8q0CAdN.o
Enter a message: Ubaid
Randomly generated key: hnnln
Encrypted message: Bontq
Decrypted message: Ubaid
```