Project Report On

Text Conversion

Submitted By: -

Vansh Sachdev 19103194

For Course: - Software Development Fundamentals-2 (SDF-2),
Course Code: - 15B11CI211

Under Guidance Of:

Assistant Prof' Dr. Alka Singhal

Department Of Computer Science

Jaypee Institute Of Information Technology
A-10, A Block, Block A, Industrial Area, Sector 62, Noida, Uttar Pradesh, 201309

- Abstract: -

Text Conversion program is a level-1-text encoding program. Which uses the asci numerical values of characters then apply some mathematical calculation to make the text in coded form later on saves the same coded information in binary files, which users can share with each other to share, which can be decoded same program as per the need of the user. Just because the size of text entered by the user may vary from very few alphabets to large paragraphs this program dynamically allocates the memory according to the need, which seems to be wastage of space-time complexity at small scale but at large scale it saves it. To Make It a Secure program this code has a part where when user enters wrong credentials 3 times the problem will delete all files related to it including itself (practically this part of code is commented out)

- General Information: -
  1. Compiler User: - gcc version 6.3.0 (MinGW.org GCC-6.3.0-1)
  2. Text Editor: - Visual Studio Code
  3. Terminal: - Gitbash

- Backend Functions: -
  1. Class UserLogin And It's Function
     a. AddUser
     b. Operator Overloading of ==
  2. **ExportUser**(UserLogin *Usser, int n): - the use of this function is to export the user information in a binary file
  3. **ImportUser**(Userlogin *UUser, int n): - the use of this function is to import the user information in a binary file
  4. **CheckFile**(): - the use of this function is to check whether user information file exist if so import the user information else create one and using predefined user information
  5. **Pushtxt**(char A): - this use of this function is to add the text
  6. **sortedInsert**(Module **head_ref, Module *new_node): - sorting function
  7. **Encryptor**(): - taking text input and encoded it
  8. **insertionSort**(Module **head_ref) sorting function
  9. **insertionSortt**(Module **head_ref, Module *new_node): -sorting function
  10. **Decryptor**(): - decoded text and print it

- Frontend Functions
  1. **main**(): - Start Whole C++ Function
  2. **LoginSwich()**: - It's for Login into the the program and use other features
  3. **Switchboard():** - it's acts as gateway into various functions of this program.

Algorithm for Encoding process

1. Start
2. Input the string
3. Get the ASCII values of each character of plain text
4. Use Mathematical formula (M*M+ (i*1000)) where m is the asci value of I Th Character.
5. Store The Calculated Value In An Array Or Linked List.
6. Repeat The Step 4 And 5 For All Elements.
7. Sort The Array Or Linked List In Ascending Order.
8. Store The Sorted Array Or Linked List In Binary File to Export The Code Along With Printing Them On Screen For User To Read The Code.

Algorithm For Decoding Process:

1. Start
2. Enter The File Name
3. Get The Numerical Value Sort The List Acc' To The Last 3 Elements Of Each Number In Array Or Linked List.
4. Take Square Root Of Each Element.
5. Save the elements into char array or linked list.(asci value being saved into char)
6. Print Array Or Linked List Elements.

- **Source Code: -**

```
/*
Project Name: - Text Convertor
Course Name: - Software Development Funcamental-2
Course Code: -15B11CI211
Course Faculty: - Dr. Alka Singhal
Made By: -
        Vansh Sachdev
        19103194
        B6

*/
#include <iostream> // Basic C++ Library
#include <vector>   // Library Used To Add Vectors DataType Of STL
#include <conio.h>  //Don't Remember Why I Used THis
//#include<bits10_1.h\stdc++.h>// Gcc Library WHich Is An Alternative To All Libraries At Onc
e
#include <fstream>   // Library For Data File Handeling
#include <algorithm> //Library For Sort Function
#include <math.h>    //Library For MatheMatical FUnctions
#include <Windows.h> //Library For System() Functions
#include <string>    //Library for string Related Wor
#include <stdio.h>   //Library For gets/puts

using namespace std;
//Prototype Of SwitchBoard Fucntion
int Switchboard();
fstream Trying("Binary.dat", ios::binary | ios::in | ios::out);

//Main Function To Drive Who Program
int main()

{
        cout.width(5);
        cout.fill(' ');
        cout << "Welcome To Text Ecryptor Software \n\n";

        Switchboard();
        cout << "\n\nCongratualations The Code Might Have Run According To Programmer's Will
\n\n";

        system("pause");
        return 0;
        Trying.close();
}
//class for UserLogin related Work
```

```
class UserLogin
{
        string Id, Password;

public:
        void AddUser(string, string);                    //class function to take data input
        friend bool operator==(UserLogin &, UserLogin &); //class friend function to overload
 equality operator
} User[3];
//defination of function used to add user
void UserLogin::AddUser(string Id, string Password)
{
        this->Id = Id;
        this->Password = Password;
}
//function used to save user info in file
void ExportUser(UserLogin *Usser, int n)
{
        fstream Trying("Binary.dat", ios::binary | ios::out);

        Trying.write((char *)Usser, n * sizeof(UserLogin));
        Trying.close();
}
// function to import saved user info from file
void importUser(UserLogin *UUser, int n)
{
        fstream Trying("Binary.dat", ios::binary | ios::in);

        Trying.read((char *)UUser, n * sizeof(UserLogin));
        Trying.close();
}

//operatort overloading function defination
bool operator==(UserLogin &FirstInfo, UserLogin &SecondInfo)
{
        return (FirstInfo.Id == SecondInfo.Id && FirstInfo.Password == SecondInfo.Password);
}
//function to check whether file user info file exist or not and if exist import data else cr
eate a new file
void CheckFIle()
{
        //this part is to check if file and exists and if exist then exctract content from it
.
        if (FILE *file = fopen("Binary.dat", "rb"))
        {
                fclose(file);

                importUser(User, 3);
```

```
        }
        else // this part is to create the file as it doesn't exist
        {
                User[0].AddUser("Vansh", "19103194");
                User[1].AddUser("Arthor", "009483");
                User[2].AddUser("1", "1");

                ExportUser(User, 3);
        }
}
//function that whill work as login window
int LoginSwitch()
{
        CheckFIle();
        string Id, Password;
        for (int j = 0; j < 3; j++)
        {
                cout << "\nEnter Login Id (trial " << j << "): - ";
                cin >> Id;
                cout << "\nEnter Password ";
                cin >> Password;
                UserLogin Trial;
                Trial.AddUser(Id, Password);

                for (int i = 0; i < 3; i++)
                {
                        if (Trial == User[i])
                        {
                                return 1;
                        }
                }
        }
        cout << "Three Attempts Passed, Now This Part Is Supposed To Remove All The Files Rel
ated To This Prgram But Because There Could Be Any Problems If This Part Messed Up So That Pa
rt Of Is Commented  And This Message Is Leaved";
        //this part will remove everything
        // remove("Binary.dat");
        // remove("Encoded.dat");
        // remove("Decoded.dat");
        // remove("tryrun.txt");
        return 0;
}
//struct that will contain and text in form of character linked list and it's code
struct Module
{
        char a;
        int b;
        Module *next;
```

```
};
//Sorting Function Bubble Sort To be
// void sortencode(int arr[], int n)
// {
//         int i, key, j;
//         for (i = 1; i < n; i++)
//         {
//                 key = arr[i];
//                 j = i - 1;

//                 while (j >= 0 && arr[j] > key)
//                 {
//                         arr[j + 1] = arr[j];
//                         j = j - 1;
//                 }
//                 arr[j + 1] = key;
//         }
// }

Module *start = NULL, *curr = NULL;
//Function To Take Text Input Word By Word
void pushtxt(char A)
{
        Module *newnode = new Module;
        newnode->a = A;
        newnode->next = NULL;
        if (start == NULL)
        {
                start = curr = newnode;
        }
        else
        {

                curr->next = newnode;
                curr = newnode;
        }
}
//Function To Sort Instered Code Module
void sortedInsert(Module **head_ref, Module *new_node)
{
        Module *current;
        /* Special case for the head end */
        if (*head_ref == NULL || (*head_ref)->b >= new_node->b)
        {
                new_node->next = *head_ref;
                *head_ref = new_node;
        }
        else
```

```
        {
                /* Locate the Module before the point of insertion */
                current = *head_ref;
                while (current->next != NULL &&
                       current->next->b < new_node->b)
                {
                        current = current->next;
                }
                new_node->next = current->next;
                current->next = new_node;
        }
}
//Function To Sort Instered Code Module

void insertionSort(Module **head_ref)
{
        Module *sorted = NULL;

        Module *current = *head_ref;
        while (current != NULL)
        {
                Module *next = current->next;

                sortedInsert(&sorted, current);

                current = next;
        }

        *head_ref = sorted;
}
//Function Where Magic Happens
void Encryptor()
{
        fstream Encoded("Encoded.dat", ios::binary | ios::out); //binary file where the Modul
e Is Stored For Shipment

        cout << "\n\nEnter The Text (Press Enter To Encode The Text Entered By You \n\n";
        char a;
        //taking Input of texty
        while ((a = getchar()) != 10)
        {
                pushtxt(a);
        }
        Module *tempa = start;
        int i = 0;
        cout << "\n\nEncrypted Version Of Text Is \n\n";
        //converting text into code and saving into same module
        while (tempa->next != NULL && tempa != NULL)
```

```
        {
                tempa->b = ((tempa->a) * (tempa->a) * 1000) + i;
                tempa = tempa->next;
                i++;
        }
        //sorting the modules
        insertionSort(&start);

        tempa = start;
        //saving the module into the files and printing them
        while (tempa != NULL)
        {
                Encoded.write((char *)tempa, sizeof(Module));
                cout << tempa->b << " ";
                tempa = tempa->next;
        }
        //closing the binary file opened above
        Encoded.close();
        //deleting the dynamic memory allocated above
        while (tempa != NULL)
        {
                Module *t = tempa;
                tempa = tempa->next;
                free(t);
        }
}
/*

Function For Converting The Code Entered By User Into Text

*/
void sortedInsertt(Module **head_ref, Module *new_node)
{
        Module *current;

        /* Special case for the head end */
        if (*head_ref == NULL || (*head_ref)->b >= new_node->b)
        {
                new_node->next = *head_ref;
                *head_ref = new_node;
        }
        else
        {
                /* Locate the Module before the point of insertion */
                current = *head_ref;
                while (current->next != NULL &&
                       current->next->b % 1000 < new_node->b % 1000)
                {
```

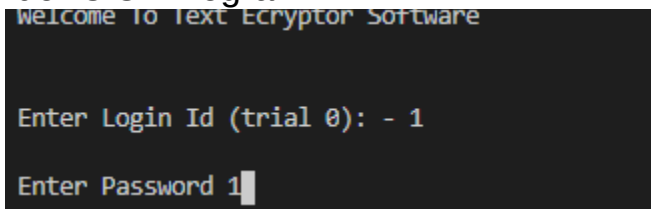
```
                        current = current->next;
                }
                new_node->next = current->next;
                current->next = new_node;
        }
}
//sorting the module for exracting the right code from the text
void insertionSortt(Module **head_ref)
{
        Module *sorted = NULL;

        Module *current = *head_ref;
        while (current != NULL)
        {
                Module *next = current->next;

                sortedInsertt(&sorted, current);

                current = next;
        }

        *head_ref = sorted;
}
//this is where the magic is rewinded
void Decryptor()
{
        fflush(stdin);
        int n = 0;

        char File[30];
        cout << "Enter the name of encrypted file(Including File Format): ";
        gets(File);
        fstream Decrypt(File, ios::binary | ios::in | ios::out); //binary file where the code
 is stored
        Module *temp = new Module;

        start = curr = temp;
        do
        {
                temp = new Module;
                Decrypt.read((char *)temp, sizeof(Module));
                curr->next = temp;
                curr = temp;
        } while (curr->next != NULL);

        cout << endl
             << endl
             << "The Code In The File: - ";
```

```
        temp = start;
        while (temp != NULL)
        {
                cout << temp->b << " ";
                temp = temp->next;
        }
        insertionSortt(&start);
        temp = start;
        while (temp != NULL)
        {
                temp->b /= 1000;
                temp = temp->next;
        }
        temp = start;
        while (temp != NULL)
        {
                temp->a = sqrt(temp->b);
                temp = temp->next;
        }
        temp = start;
        cout << endl
             << endl;
        while (temp != NULL)
        {
                cout << temp->a;
                temp = temp->next;
        }
        Decrypt.close();
}
//Switch Board Of Program Function That's Runing the Entire Program
int Switchboard()
{
        if (LoginSwitch())
        {
                system("clear");
                cout << "Login Succesfull \n\n";

                system("pause");
                system("clear");
                cout << "Welcome User,\n You Have Following Options To Choose From\n\nPress 1
 For Encrypting New Text\n\nPress 2 For Decrypting The Code\n\nPress 5 For Developer's Inform
ation\n\nPress 6 For Exiting The Function\n\n";
                int option;

                scanf("%d", &option);
                fflush(stdin);
                if (option == 1)
```

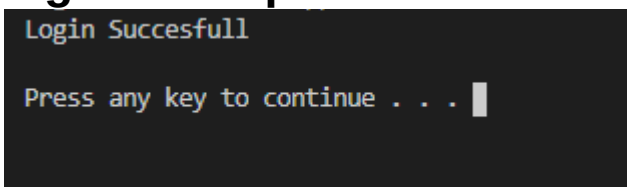
```
                        Encryptor();
                else if (option == 2)
                        Decryptor();

                else if (option == 6 || option == 5)
                        system("pause");
                else if (option == 9483)
                {
                        cout << "Easter Egg: - \n\nDeleting All Encryptions And Decryptions h
ehehe ";
                        remove("Encoded.dat");
                        remove("Decoded.dat");
                }

                return 0;
        }
}
        /*

Good Bye
Project Code Documentation Part End
Written By: -
        Vansh Sachdeva
        19103194
        B6

*/
```

Output: -

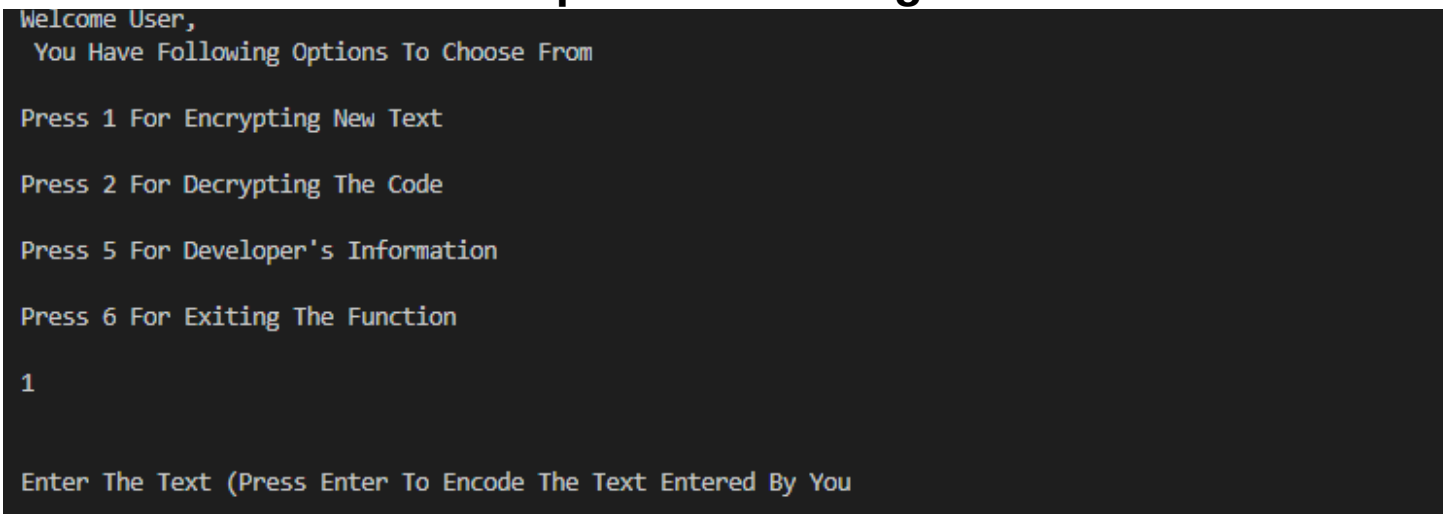
Welcome And Login Windows Of Program: -



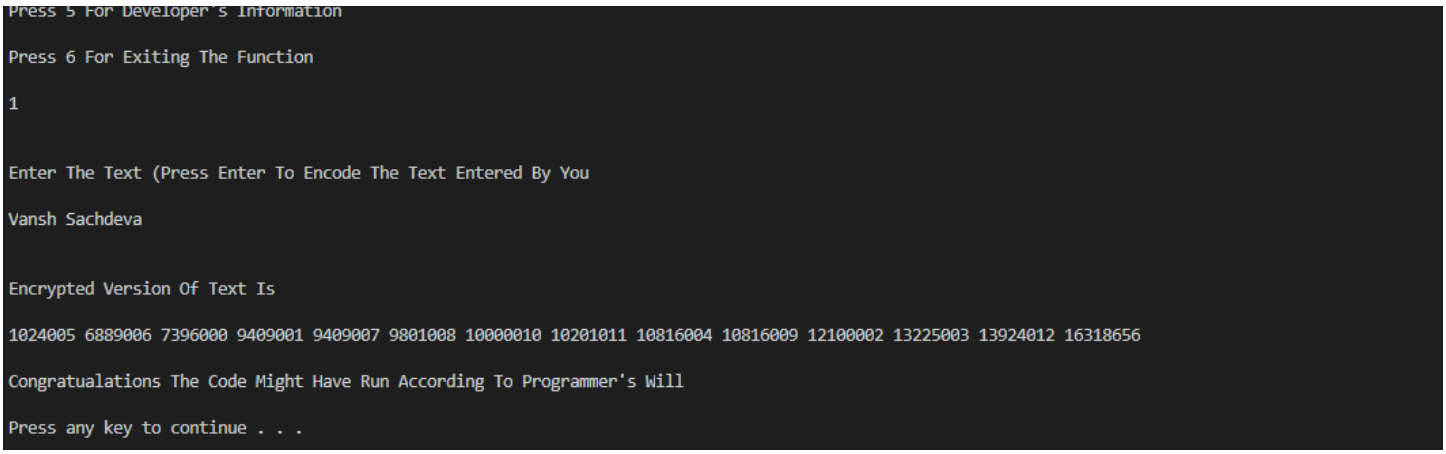
After Successful Login Attempt: -



SwitchBoard Which Is Opened After Login: -



After Selecting First Option Which Is of Encoding Text and Trying a Sample text;



After Selecting Second Option Which Is Of Decoding The Code In Binary File And Printing The Text With Some Garbage.

```
Welcome User,
 You Have Following Options To Choose From

Press 1 For Encrypting New Text

Press 2 For Decrypting The Code

Press 5 For Developer's Information

Press 6 For Exiting The Function

2
Enter the name of encrypted file(Including File Format): encoded.dat


The Code In The File: - 7798976 1024005 6889006 7396000 9409001 9409007 9801008 10000010 10201011 10816004 10816009 12100002 13225003 13924012 16318656

 VanshSachdevX

Congratualations The Code Might Have Run According To Programmer's Will

Press any key to continue . . .
```

Files Created By Code During Runtime (First Time Also)