

C Programming – Final Project

By: Chanmeet Singh

Prologue: Encoding and decoding play a vital role in modern communication and data storage. With the ever-increasing amount of data being transmitted and stored, the need for efficient and secure methods of encoding and decoding is more pressing than ever. Encoding and decoding is a method of representing data in a different format to efficiently transfer information through a network or the web. The encoder converts data into a web representation. Once received, the decoder converts the web representation data into its original format

Aim: The aim of this project is to provide a simple and flexible solution for encoding and decoding any inputted string or contents of a file. This can be used for various purposes such as securing sensitive information, or transforming data into a format that is easier to transmit or store.

Idea: The project uses a basic three-level operation to encode and decode the input, providing a simple and fast method of transforming data. The program is written in C and is designed to be easy to use and understand, making it suitable for a wide range of applications. The project first converts the given contents to their ASCII value and then this value is converted to a random base number system with all the numerical digits converted to symbols.

Program:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <conio.h>

int generate_random_base() {
    srand(time(0));
    int base = (rand() % (10 - 2 + 1)) + 2;
    return base;
}

int BaseToAscii (long int number, int base) {
    long int temp = number;
    int ascii_value = 0;
    int temp_count = 0;
    while (temp!=0) {
        int last_digit = temp%10 ;
        ascii_value += (last_digit * pow(base,temp_count));
        ++temp_count;
        temp = temp/10;
    }
}
```

```

    }
    return ascii_value;
}

void delay(int number_of_seconds) {
    int milli_seconds = 1000 * number_of_seconds;
    clock_t start_time = clock();
    while (clock() < start_time + milli_seconds)
        ;
}

int main() {

    char test_string [256];
    printf("\nEnter the string to be encrypted: ");
    fgets(test_string, 255, stdin);
    int n = strlen(test_string);
    int encryption_base = generate_random_base() ;

    printf("\nYour Key is %d. Please remember it otherwise you can't decrypt your
message.\n", encryption_base);
    printf("\nEncrypting..\n");
    delay(3);

    int final_length = n*16;
    int encrypted_msg [final_length] ;
    int pos=0;

    for (int i=0; i<n; ++i) {
        char c = test_string[i];
        int c_ascii = (int)c;
        int reverse_converted [16];
        int temp = c_ascii;
        int n2=0;
        int subscript = 0;
        while (temp>0) {
            reverse_converted[subscript] = (temp%encryption_base);
            temp = temp/encryption_base;
            n2++;
            ++subscript;
        }

        int final_converted [n2];
        int k=0;

        for (int j = n2 - 1; j >= 0; --j) {
            final_converted[k] = reverse_converted[j];
            ++k;
        }
        for (int l=0; l<n2; ++l) {
            encrypted_msg[pos+l]=final_converted[l];
        }

        pos+=n2;
        encrypted_msg[pos] = 9999;
    }
}

```

```

    pos+=1;
}

char super_encrypted_msg [pos];

for (int g=0; g<pos; ++g) {

    if (encrypted_msg[g] == 0)
        super_encrypted_msg[g] = ')' ;
    else if (encrypted_msg[g] == 1)
        super_encrypted_msg[g] = '!' ;
    else if (encrypted_msg[g] == 2)
        super_encrypted_msg[g] = '@' ;
    else if (encrypted_msg[g] == 3)
        super_encrypted_msg[g] = '#' ;
    else if (encrypted_msg[g] == 4)
        super_encrypted_msg[g] = '$' ;
    else if (encrypted_msg[g] == 5)
        super_encrypted_msg[g] = '%' ;
    else if (encrypted_msg[g] == 6)
        super_encrypted_msg[g] = '^' ;
    else if (encrypted_msg[g] == 7)
        super_encrypted_msg[g] = '&' ;
    else if (encrypted_msg[g] == 8)
        super_encrypted_msg[g] = '*' ;
    else if (encrypted_msg[g] == 9)
        super_encrypted_msg[g] = '(' ;
    else if (encrypted_msg[g] == 9999)
        super_encrypted_msg[g] = '~' ;
}

printf("\nYour encrypted message is:\n");
for (int m=0; m<pos; ++m) {
    printf("%c ",super_encrypted_msg[m]);
}

printf("\n\nPress Enter to continue with decryption ");
char temp_choice;
scanf("%c",&temp_choice);

int key ;
printf("\n\nEnter your Decryption Key: ");
scanf("%d",&key);
if (key != encryption_base) {
    printf("\nYou entered the wrong Key. You have only one chance left. Enter Again: ");
    scanf("%d",&key);
    if (key != encryption_base) {
        printf("\nSorry you again entered the wrong key. You can't decrypt your
message.");
        exit(1);
    }
}
}

```

```

// DECRYPTION :

char temp_encrypted [final_length];
for (int p=0; p<pos; ++p) {
    temp_encrypted[p] = super_encrypted_msg[p];
}
printf("\n\n\nDecrypting...");
delay(3);

int first_decrypted [pos];
for (int q=0; q<pos; ++q) {
    if (temp_encrypted[q] == ')')
        first_decrypted[q] = 0;
    else if (temp_encrypted[q] == '!')
        first_decrypted[q] = 1;
    else if (temp_encrypted[q] == '@')
        first_decrypted[q] = 2;
    else if (temp_encrypted[q] == '#')
        first_decrypted[q] = 3;
    else if (temp_encrypted[q] == '$')
        first_decrypted[q] = 4;
    else if (temp_encrypted[q] == '%')
        first_decrypted[q] = 5;
    else if (temp_encrypted[q] == '^')
        first_decrypted[q] = 6;
    else if (temp_encrypted[q] == '&')
        first_decrypted[q] = 7;
    else if (temp_encrypted[q] == '*')
        first_decrypted[q] = 8;
    else if (temp_encrypted[q] == '(')
        first_decrypted[q] = 9;
    else if (temp_encrypted[q] == '~')
        first_decrypted[q] = 9999;
}

printf("\n\n");
for (int r=0; r<pos; ++r) {
    printf("%d ",first_decrypted[r]);
}
delay(1);

int second_decrypted [n];
int second_decrypted_index = 0;
int previous_9999_index = -1;
for (int s=0; s<pos; ++s) {
    if (first_decrypted[s] == 9999) {
        int current_9999_index = s;
        int len_of_s;
        if (previous_9999_index != -1) {
            len_of_s = current_9999_index - previous_9999_index - 1;
        } else {
            len_of_s = current_9999_index;

```

```

    }
    char temp_str [len_of_s];
    int temp_i=0;
    for (int t=previous_9999_index+1; t<current_9999_index; ++t) {
        char temp_char = first_decrypted[t]+'0';
        temp_str[temp_i] = temp_char;
        ++temp_i;
    }
    int base_int_value = atoi(temp_str);
    second_decrypted[second_decrypted_index] = base_int_value;
    ++second_decrypted_index;
    previous_9999_index = current_9999_index;
}

}

printf("\n");
for (int u=0; u<n; ++u) {
    printf("%d ",second_decrypted[u]);
}
delay(1);

int third_decrypted [n];
for (int v=0; v<n; ++v) {
    third_decrypted[v] = BaseToAscii(second_decrypted[v],encryption_base);
}
printf("\n");
for (int w=0; w<n; ++w) {
    printf("%d ",third_decrypted[w]);
}
delay(1);

char final_decrypted [n-1];
for (int z=0; z<n-1; ++z) {
    char temp_c = third_decrypted[z];
    final_decrypted[z] = temp_c;
}

printf("\n\nYour DECRYPTED message is : ");
for (int y=0; y<n-1; ++y) {
    printf("%c",final_decrypted[y]);
}
printf("\n\n");

return 0;
}

```

Outputs:

```
PS C:\Users\chanm\OneDrive\Desktop\C Project Clean> ./main.exe
```

```
Enter the string to be encrypted: Hello_World
```

```
Your Key is 7. Please remember it otherwise you can't decrypt your message.
```

```
Encrypting..
```

```
Your encrypted message is:
```

```
! # @ ~ @ ) # ~ @ ! # ~ @ ! # ~ @ ! ^ ~ ! ^ $ ~ ! % # ~ @ ! ^ ~ @ @ @ ~ @ ! # ~ @ ) @ ~ ! # ~
```

```
Press Enter to continue with decryption 7
```

```
Enter your Decryption Key: 7
```

```
Decrypting...
```

```
1 3 2 9999 2 0 3 9999 2 1 3 9999 2 1 3 9999 2 1 6 9999 1 6 4 9999 1 5 3 9999 2 1 6 9999 2 2 2 9999 2 1 3 9999 2 0 2 9999 1 3 9999
132 203 213 213 216 164 153 216 222 213 202 132
72 101 108 108 111 95 87 111 114 108 100 72
```

```
Your DECRYPTED message is : Hello_World
```

```
PS C:\Users\chanm\OneDrive\Desktop\C Project Clean> ./main.exe
```

```
Enter the string to be encrypted: ishaan69420
```

```
Your Key is 6. Please remember it otherwise you can't decrypt your message.
```

```
Encrypting..
```

```
Your encrypted message is:
```

```
@ % # ~ # ! ! ~ @ % @ ~ @ $ ! ~ @ $ ! ~ # ) @ ~ ! # ) ~ ! # # ~ ! @ $ ~ ! @ @ ~ ! @ ) ~ ! $ ~
```

```
Press Enter to continue with decryption
```

```
Enter your Decryption Key: 6
```

```
Decrypting...
```

```
2 5 3 9999 3 1 1 9999 2 5 2 9999 2 4 1 9999 2 4 1 9999 3 0 2 9999 1 3 0 9999 1 3 3 9999 1 2 4 9999 1 2 2 9999 1 2 0 9999 1 4 9999
253 311 252 241 241 302 130 133 124 122 120 140
105 115 104 97 97 110 54 57 52 50 48 60
```

```
Your DECRYPTED message is : ishaan69420
```

```
PS C:\Users\chanm\OneDrive\Desktop\C Project Clean> ./main.exe
```

```
Enter the string to be encrypted: projectdone
```

```
Your Key is 6. Please remember it otherwise you can't decrypt your message.
```

```
Encrypting..
```

```
Your encrypted message is:
```

```
# ) $ ~ # ! ) ~ # ) # ~ @ % $ ~ @ $ % ~ @ $ # ~ # ! @ ~ @ $ $ ~ # ) # ~ # ) @ ~ @ $ % ~ ! $ ~
```

```
Press Enter to continue with decryption 10
```

```
Enter your Decryption Key:
```

```
You entered the wrong Key. You have only one chance left. Enter Again: 1
```

```
Sorry you again entered the wrong key. You can't decrypt your message.
```

Conclusion: In conclusion, this project provides a useful tool for encoding and decoding data in today's fast-paced digital world, offering a simple and secure solution for protecting sensitive information and transforming data for more efficient storage and transmission.

Thanks

Team Members: **Chanmeet Singh** – 08 – K1 – CSE
 Ishaan Minocha – 31 – K1 – CSE
 Haaris Huda – 13 – K1 – CSE
 Rishi Agarwal – 48 – K2 – CSE
