CST 1500

AISHA IDOO

SHARON SALAMI
M00940322

MORAANUOLUWA AKINBINU
M00858828

17TH APRIL 2023

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <stdbool.h>
#include <unistd.h>

#define FILENAME_SIZE 1024
#define MAX_LINE 2048

int randomize(int a, int b);
void fill_unique(int array[], int length, int min, int max);
int main(){
int rand_num, words;
	printf("Number of words: ");
	scanf("%d", &words);  //prompts the user to input  length of password
	while(true){
		if (words > 2 && words < 5){
			puts("You can now proceed"); // Allows the user to proceed once the no of words is within valid parameters
			break;
	}
```

```
    else{
        puts("You have entered an invalid parameter.");
        printf("Number of words: ");
        scanf("%d", &words);
    }
    }
    printf("Generating Password\n");
  FILE* dict;  //pointer dict points to the dictionary file
  char filename[FILENAME_SIZE];
  char buffer[MAX_LINE];
  int read_line, read_file;
  int myvar, i, index = 1 ;

  srand (time(NULL) * getpid());

  int myarray[words];  // creates an array with the max no of elements to be the no of words entered by the user.
  fill_unique(myarray, words, 1, words);// e.g [2, 3, 1, 4]
  myvar = randomize(200, 1);
  read_line = myvar; // randomly generate the line to read from the file.....This number will remain constant i.e the number generated will print out words from the same line in different files.
```

```
   char word1[10];
   char word2[10];
   char word3[10];
   char word4[10];
    for (i = 1; i <= words; i++)
    {
       if (i == 1 )
       {
          read_file = myarray[0];
       }
       else if (i == 2)
       {
          read_file = myarray[1];
       }
       else if (i == 3)
       {
          read_file = myarray[2];
       }
       else if (i == 4)
       {
          read_file = myarray[3];
       }
// picking the dictionary files at random
```

```
        // dictionary 1
        if (read_file == 1)
        {
            dict = fopen("C:\\Users\\moraa\\Desktop\\cwfiles\\4",
"r");
            if (dict == NULL){
                printf("Unable to open file");
                return 1;
            }

            bool keep_reading = true;
            int current_line = 1;

            do{
                fgets(buffer, MAX_LINE, dict);
                if (feof(dict)){
                    keep_reading = false;
                }
                else if (current_line == read_line){
                    keep_reading = false;
                    printf("-> %d ", index);
                    index++;
                    puts(buffer);
```

```
            }
            current_line++;
        }while(keep_reading);
        fclose(dict);
    }
    // dictionary 2
    if (read_file == 2)
    {
        dict = fopen("C:\\Users\\moraa\\Desktop\\cwfiles\\5",
"r");
        if (dict == NULL){
            printf("Unable to open file");
            return 1;
        }
        bool keep_reading = true;
        int current_line = 1;
        do{
            fgets(buffer, MAX_LINE, dict);
            if (feof(dict)){
                keep_reading = false;
            }
            else if (current_line == read_line){
                keep_reading = false;
```

```
                printf("-> %d ", index);
                index++;
                puts(buffer);
            }
            current_line++;
        }while(keep_reading);
        fclose(dict);
        }
    // dictionary 3
    if (read_file == 3)
    {
        dict = fopen("C:\\Users\\moraa\\Desktop\\cwfiles\\6",
"r");
        if (dict == NULL){
            printf("Unable to open file");
            return 1;
        }
        bool keep_reading = true;
        int current_line = 1;
        do{
            fgets(buffer, MAX_LINE, dict);
            if (feof(dict)){
                keep_reading = false;
```

```
            }
            else if (current_line == read_line){
                keep_reading = false;
                printf("-> %d ", index);
                index++;
                puts(buffer);
            }
            current_line++;
        }while(keep_reading);
        fclose(dict);
    }
    // dictionary 4
    if (read_file == 4)
    {
        dict = fopen("C:\\Users\\moraa\\Desktop\\cwfiles\\7",
"r");
        if (dict == NULL){
            printf("Unable to open file");
            return 1;
        }
        bool keep_reading = true;
        int current_line = 1;
        do{
```

```
            fgets(buffer, MAX_LINE, dict);
            if (feof(dict)){
                keep_reading = false;
            }
            else if (current_line == read_line){
                keep_reading = false;
                printf("-> %d ", index);
                index++;
                puts(buffer);
            }
            current_line++;
        }while(keep_reading);
        fclose(dict);

// The code below catches the word per the loop's iteration
    }
    if (i == 1)
    {
        strcpy(word1, buffer);
    }
    else if (i == 2)
    {
        strcpy(word2, buffer);
```

```
        }
        else if (i == 3)
        {
            strcpy(word3, buffer);
        }
        else if (i == 4)
        {
            strcpy(word4, buffer);
        }
    }
    if (words == 3)
    {
        printf("Your e-t-r password is: %s-%s-%s", word1,
word2, word3);


    }
    else
    {
        printf("Your e-t-r password is \a%s-%s-%s-%s", word1,
word2, word3, word4);
    }
```

```
    return(0);
}

// functions
    // function to generate a random number
int randomize(int a, int b){
        int rand_num1;
        srand (time(NULL) * getpid());
        rand_num1 = rand() % a + b;
    return rand_num1;
    }
// this generates random number and stores them uniquely in an array to be accessed.
void fill_unique(int array[], int length, int min, int max)
{
    int new_random;
    bool unique;
    for(int i = 0; i < length; i++)
    {
        do
        {
            new_random = (rand() % (max - min + 1)) + min;
            unique = true;
```

```
            for (int j = 0; j < i; j++)
            {
                if (array[j] == new_random) unique = false;
            }
        }while(!unique);
        array[i] = new_random;
    }
}
```

PROJECT TITLE

PASSWORD GENERATOR

AUTHORS: SHARON TELANGNI SALAMI & AKINBINU MORAANUOLUWA

PROJECT DESCRIPTION
==========

This program is a password generator that is written in C languange and
compiled using the gcc compiler (on development of the code). We encountered
some problems like learning C from an extra source to get a better understanding
of how the programming language works. We included functions to make sure our code is less clumpy.

Instructions
==========
To run this code, you will need to compile it using a C compiler like GCC. As the
code is terminal base, you will get an output on the command prompt.

Functions included (2)
==========
As the program is not a complex code, we only used two functions which are:

1. The randomize () function which takes two parameters is used to generate
random numbers and we implemented this when iterating through the words in
the dictionary files.

2. The program uses the fill unique function to generate an array of unique
random numbers between 1 and the length of the password.
These numbers are used to determine which dictionary file to read.
The program reads the selected
line from the dictionary file and stores the word in a buffer.

```
moraanu@moraanu-VirtualBox:~/Desktop/cst1500/coursework$ ./'Our Coursework'
Number of words: 3
You can now proceed
Generating Password
-> 1 advice

-> 2 after

-> 3 back

Your e-t-r password is: advice
-after
-back
moraanu@moraanu-VirtualBox:~/Desktop/cst1500/coursework$
```

```
moraanu@moraanu-VirtualBox:~/Desktop/cst1500/coursework$ ./'Our Coursework'
Number of words: 4
You can now proceed
Generating Password
-> 1 alarm

-> 2 airline

-> 3 agency

-> 4 bath

Your e-t-r password is alarm
-airline
-agency
-bath
moraanu@moraanu-VirtualBox:~/Desktop/cst1500/coursework$
```