ADJUSTING TO THE NEW NORMAL

A Final Project
Presented to the Faculty of Engineering
De La Salle University-Manila
3rd Term, A.Y. 2019-2020

In partial fulfillment
of the requirements for the course
COMPUTER FUNDAMENTALS AND PROGRAMMING 2
LBYEC2B

Bautista, Margarita
Mayuga, Keenan
Section EQ1

Mr. Jack Catalan

October 2, 2020

I. Introduction

Since the beginning of the COVID-19 and quarantine, everything has been transferring to the digital world wherein almost everything needs to be typed. Thus, many people are adjusting to the new normal through digital means. In relation to this, having the speed and accuracy in typing will be of great advantage especially at this time. It can save time, decrease fatigue, improve posture and prevent injuries such as repetitive stress injuries (RSI), and increase productivity (SyberScribe, 2019). With this, a typing game would be beneficial for the users to be aware of their current speed and accuracy and would be given a chance to improve them. Given that this type writing program is also in the form of a game, it can also relieve the stress of work from home, online classes, and all other causes of stress in this time of the pandemic. This typing game was programmed using Object Oriented Programming in C programming.

II. Classes and Objects

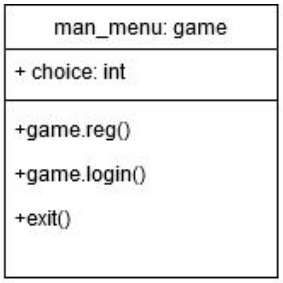
A. Class

The class Type has attributes such as the fname, lname, username, password, tests, num, and next that will be used throughout the program. The reg method allows the user to register by inputting one's first and last name as well as one's username and password. After registration, a text file with the username as the filename is created in the same folder as the program. The first and last name as well as the password will be put in the text file. For the login method, the username and password inputted by the user will be verified if it is registered. The myprofile method shows the records of the user. The listTest method shows the set of texts that the user can choose from. The set of text that

the user will choose will be the one to be typed during the game. The Test method is the typing game itself where the user will type the text and see the results of the game. Lastly, the Statistics method shows the statistics of the mistyped words.

B. Objects

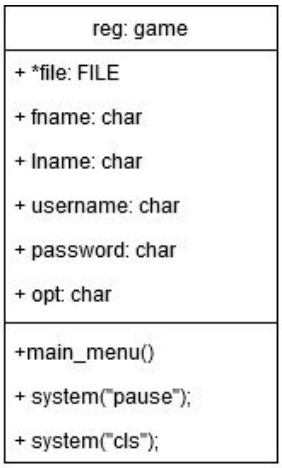
a. main_menu()



The main_menu object displays the main menu and asks the user to input a choice.

(1) Register
(2) Log In
(3) Exit

b. reg()



The reg object asks the user to input their First name, Last name, Username, and Password. These are saved in the FILE file so that the user can exit the game but still save their progress. Once finished registering, the program will go back to main_menu.

c. login()

login: game
+ *file: FILE + tmp[20]: char + user: char + opt: char + pass: char
+main_menu() +usermenu(char *user) + system("pause"); + system("cls");

Login object is where the user will input the username and password after registering. The user's input will be checked from the saved user files. If the login was successful then the program will call the usermenu object. If not, then it will go back to main_menu.

d. usermenu(char *user)

usermenu(char *user): game
+ f: FILE + choice: int + user: char
+game.listTest(char *user) +game.myprofile(char *user) +game.Statistics(char *user) +main_menu +exit() + system("cls"); + system("pause");

Usermenu object displays another menu that is for the user and the user must input a choice.

(1) Start
(2) My Profile
(3) Statistic
(4) Logout
(5) Exit

e. myprofile(char *user)

myprofile(char *user) : game
+ f: FILE + fname: char + lname: char + username: char + ch: char
+ system("pause"); + system("cls");

Myprofile object displays the profile of the user by opening the user's file. It shows the first name, last name, username. It also displays the user's previous test records which contain wrong words, accuracy, test time, and word per minute(wpm).

f. listTest(char *user)

listTest(char *): game
+ f: FILE + user: char + ch: char + newnode: Type + new: Type + last: *Type + start: *Type + ptr: *Type + count: int + num: int + tests[30]: char
+ Test(char *value, char *user) + usermenu(char *user) + system("pause"); + system("cls");

listTest object asks the users which word selection he/she to choose. Each selection is 10 lines each containing a sentence taken from Harvard Sentences, which are a collection of words which uses specific syllables for voice standardization. It is also a good collection of words for typing tests standardization and is used in some online word per minute tests.

g. Test(char *value, char* user)

Test(char *value, char*user)
+ f: FILE + f1: FILE + count: int + wrong: int + line: int + ch: char + text[2000]: char + f10: FILE + f20: FILE + *wrongfile: FILE + s1: string + s2: string + file1: string + file1: string + wrongwords: string + buf[2000]: char + c1: float + accuracy: float + time: float + exact: float + speed: int
+ clock() + system("pause"); + system("cls");

The Test object is the game/run itself. It first displays the chosen word selection file then asks the user how many lines (sentences) he/she wants to type for the test. Once the user starts typing a timer is started. When the user's finish, the timer ends and the input will be printed into a check.txt file. Each word is then strcmp to the word selection file where if it is equal then correct word count will increase by 1, if it is not equal to then wrong words count will increase by 1 and will be added to a wrong.txt file. Then the accuracy and word per minute will be calculated and displayed along with the time, which is also appended to your user's file record.

h. Statistics(char*)

Statistic(char *user): game
+ *stats: FILE + count: int + index: int + SIZE = 2000: int + word[SIZE]: int + c: char + *ctr: char
+ system("pause"); + system("cls");

The Statistics object displays the mistyped words through the wrong.txt file. Then it asks the user to input a word that will be counted and displayed how many times it shows up. Thus showing how many times the word was mistyped.

III. Program Walkthrough

A.Main Menu

The first thing that will appear on the screen will be three options that the user can choose from. These choices include register, log in, and exit.

B. Register

```
 ____ ____ ____ ____ ____ ____ ____ ____
|__  |  __|  __|    |  __|_   _|  __| __ |
|  -|  __|  |  |-   -|__  | | |  __|  -|
|__|__|____|____|____|____| |_| |____|__|__|

*************************** TYPING TEST ******************************

Enter First name: Margarita

Enter last name: Bautista

Enter Username: LBYEC2B

Enter password : Program

*************************** TYPING TEST ******************************
Press any key to continue . . .
```

If the user chooses 1, the register window will appear. The user will be asked to enter his or her first name, last name, username, and password.

```
LBYEC2B - Notepad
File  Edit  Format  View  Help
Margarita
Bautista
Program
```

After registration, a text file with the username as the filename will be created. This will consist of the first name, last name, as well as the password.

```
 ____ ____ ____ ____ ____ ____ ____ ____
| __ |  __|  __|    |  __|_  _|  __| __ |
|  -|  __|  |__|-  -|__   | | |  __|  -|
|__|__|____|____|____|____| |_| |____|__|

*************************** TYPING TEST ******************************

Enter First name: Keenan

Enter last name: Mayuga

Enter Username: LBYEC2B

 User Already Exist

 Press X to return to main menu or Y to try again: Y

Enter Username: KM

Enter password : Prog

*************************** TYPING TEST ******************************
Press any key to continue . . .
```

If the user attempts to register with the same username and password, a message will appear telling the user that the user already exists. The user can choose from trying again or going back to the main menu.

C. Login

```
 __   ____ ____   ____ ____
|  |  |    |  __|  |    |  | |
|  |__|  |  |  |  |-  -| | | |
|____|____|____|  |____|_|__|

*************************** TYPING TEST ******************************

Enter Username : LBYEC2B

Enter Password : Program

 Log In Successful!
Press any key to continue . . .
```

If the user chooses 2, the login screen will appear wherein the user will be asked to enter the username and password. If the username and password is verified a message will appear stating that the login is successful.

```
 __   ____ ____   ____ ____
|  |  |    |  __|  |    |  |
|  |__|  |  |  |  |-   -| | |
|____|____|____|  |____|_|__|

**************************** TYPING TEST ******************************

Enter Username : Keenan

 User does not exist!

 Press X to return to main menu or Y to try again: Y

Enter Username : KM

Enter Password : Program

 Wrong Password

 Press X to return to main menu or Y to try again: Y

Enter Password : Prog

 Log In Successful!
Press any key to continue . . .
```

An error message will show if the password inputted is wrong. The user can choose from trying again or going back to the main menu.

```
 __   ____ ____   ____ ____
|  |  |    |  __|  |    |  |
|  |__|  |  |  |  |-   -| | |
|____|____|____|  |____|_|__|

**************************** TYPING TEST ******************************

Enter Username : lby

 User does not exist!

 Press X to return to main menu or any other key to try again:
```

The program can also detect if the username inputted is not registered and will show the error message that the user does not exist.

D. Game Menu

```
********************************** TYPING TEST ********************************
============================== MENU ==================================
(1) Start
(2) My Profile
(3) Statistics
(4) Logout
(5) Exit
Enter Choice:
```

Once the login is successful, another screen will appear showing 5 choices that the user can choose from. The choices are starting the game, viewing the user's profile, viewing the statistics of the game, logging out, or exiting the program.

E. Game Proper

```
********************************** TYPING TEST ********************************

(0)Selection 0
(1)Selection 1
(2)Selection 2
(3)Selection 3

 Choose Selection 3
```

When the user chooses 1, the game will start by asking the user which set of text to type.

```
**************************** TYPING TEST ******************************
The slush lay deep along the street.
A wisp of cloud hung in the blue air.
A pound of sugar costs more than eggs.
The fin was sharp and cut the clear water.
The play seems dull and quite stupid.
Bail the boat, to stop it from sinking.
The term ended in late June that year.
A tusk is used to make costly gifts.
Ten pins were set in order.
The bill as paid every third week.

How many lines do you want to write of above selection: 1

Start Writing

Press any key to continue . . .
The slush lay deep along the street

 No of words correct : 6
 No of words Incorrect : 1
 Accuracy is : -536870912 percent
 Typing Time : 8.000000
 Speed : 44 WPM
**************************** TYPING TEST ******************************
Press any key to continue . . .
```

The chosen set of text will appear on the screen. The user will be asked as to how many lines he or she wants to type. After typing, the results of the test will appear. This consists of the number of correct words, number of incorrect words, accuracy, typing time, and speed. After the game, the user will be returned to the game menu, where he or she can choose to play again, view the profile, view the statistics, logout, or exit the program.

```
 ____ ____ ____ ____ ____ __   ____
|  _ | __ |    |   __|    |  |  |   __|
|   __|    -|  |  |   __|-   -|  |__|   __|
|__|  |__|__|____|__|  |____|____|____|


**************************** TYPING TEST ******************************

 Name : Margarita
Bautista
Program
This test result correct words : 6 Wrong words : 1 Accuracy : 85.7143 Test Time : 8 Speed : 44 WPM Last Name: rogram
This test result correct words : 6 Wrong words : 1 Accuracy : 85.7143 Test Time : 8 Speed : 44 WPM
 User name : ult correct words : 6 Wrong words : 1 Accuracy : 85.7143 Test Time : 8 Speed : 44 WPM
 Your Record

 Press any key to exit Press any key to continue . . .
```

If the user chooses 2 in the game menu, the test result will appear on the screen.

```
********************************** TYPING TEST ********************************
        MISTYPED WORDS
canoo smoth drak bloo backround
Enter one of the words above to see how many times it is mistyped: canoe
The word canoo smoth drak bloo backround is mistyped 7468540 timesPress any key to continue . . .
```

If the user chooses 3 in the game menu, the statistics of the mistyped words will appear.

```
********************************** TYPING TEST ********************************
============================ LOG IN ==================================
(1) Register
(2) Log In
(3) Exit
Enter Choice:
```

If the user chooses to logout, it goes back to the main menu where the user can register, login, or exit the program.

```
********************************** TYPING TEST ********************************
========================== LOG IN ==================================
(1) Register
(2) Log In
(3) Exit
Enter Choice:
--------------------------------
Process exited after 1198 seconds with return value 0
Press any key to continue . . .
```

If the exit option is chosen, the screen will appear similar to above. If any key is pressed, the window will close.

```
********************************** TYPING TEST ********************************
============================ MENU ====================================
(1) Start
(2) My Profile
(3) Statistics
(4) Logout
(5) Exit
Enter Choice:
--------------------------------
Process exited after 73.26 seconds with return value 0
Press any key to continue . . .
```

The exit in the game menu is similar to the exit of the main menu wherein the if any key is pressed, the window will close.

IV. Code

A. main.cpp

```
#include "typing.h"
```

```
int main()
{

   main_menu();
   return 0;
}
```

B. typing.h

```
#ifndef TYPING_H
#define TYPING_H

#include <iostream>
#include <conio.h>
#include <stdlib.h>
#include <fstream>
#include <string.h>
#include <time.h>

using namespace std;

class Type
{
   char fname[20], lname[20], username[20], password[20];

   public:
          char tests[30];
       int num;
          Type *next;
          void usermenu(char*);
          void reg();
          void login();
          void listTest(char*);
          void myprofile(char*);
          void Test(char*, char*);
          void Statistics(char*);

};
```

```
void main_menu();




#endif
```

C. typing.cpp

```
// WRITTEN BY:  Keenan Mayuga and Margarita Bautista
// SECTION: EQ1
//
// FOR COURSE:      LBYEC2B
//
// PURPOSE:
// This program assess the typing speed and accuracy of the user.
//
// OVERALL METHOD/ALGORITHM:
// The list of general tasks is:
// 1. Display a main menu and ask the user to select.
// 2. Execute the menu item.
//        2-1. If selected menu item is '1', perform register
//        2-1-1. Asks the user to enter first name, last name, username, and password
//        2-1-2. If the registration is successful, the main menu will be shown again.
//        2-1-3. If the the user attempts to register with the same username and password,
//                a message will appear telling the user that the user already exists and is ask
//                whether to go back to the main menu or try again
//                2-1-3-1. If selected menu item is 'x' or 'X', display main menu
//                2-1-3-2. If selected menu item is 'y' or 'Y', asks for a username again.
//        2-2. If selected menu item is '2', perform login
//        2-2-1. If the login is successful, display usermenu to select.
//                2-2-1-1. If selected menu item is '1', display text selection menu and ask the user
to select
//                2-2-1-1-1. Display the chosen text
//                2-2-1-1-2. Ask the user the number of lines to be typed
//                2-2-1-1-3. Ask the user to start typing
```

```
//                  2-2-1-1-4. Display the results (number of words correct, number of words
incorrect,
//                          accuracy, typing time, and speed)
//                  2-2-1-2. If selected menu item is '2', display the profile of the user
//                  2-2-1-3. If selected menu item is '3', display the statistics of mistyped words
//                  2-2-1-4. If selected menu item is '4', logout and display main menu
//                  2-2-1-5. If selected menu item is '5', exit the program
//          2-2-2. If the username is not verified, a message will appear
//                  telling the user that the user does not exists and is asked whether to go back to
main menu
//                  or try again
//                  2-2-2-1. If selected menu item is 'x' or 'X', display main menu
//                  2-2-2-2. If selected menu item is 'y' or 'Y', asks for a username again
//          2-2-3. If the password is incorrect, a message will appear telling the user that the
password is incorrect
//                  is asked whether to go back to main menu or try again
//                  2-2-3-1. If selected menu item is 'x' or 'X', display main menu
//                  2-2-3-2. If selected menu item is 'y' or 'Y', asks for a password again
//
//          2-3. If selected menu item is '3', exit the program
//
// CREDITS:
//          https://github.com/iamsahil1910Typing-Test
//
// FUNCTIONS:
//
// main_menu
//          display the menu
//
// reg
//          ask user to register
//
// login
//          ask user to login
//
// usermenu
//          display the menu for the game
//
// myprofile
```

```
//          displays the past records of the user
//
// listTest
//          display menu for text selection
//
// Test
//          executes the typing game and shows the result
//
// Statistics
//          displays statistics of mistyped words
//
// INCLUDED FILES:
//          typing.h
//
// DATA FILES:
//          username.txt
//          check.txt
//          wrong.txt
#include "typing.h"

Type game;

Type *start = NULL, *last = NULL;

// FUNCTION NAME: main_menu
// CREDITS:
//
// PARAMETERS: None
//
// RETURN VALUE: None
//
// CALLS TO: none
//
// CALLED FROM: main()
//
// METHOD:
//
// This function displays the main menu
void main_menu()
```

```
{
   int choice;//menu key variable
   do
   {
         system("cls");
         printf("\t        __       _____ _____   _____ _____ \n");
         printf("\t        | | |     |  __| |            |  | |\n");
         printf("\t        | |__| | | | | |-   -| | | |\n");
         printf("\t        |_____|_____|_____|  |_____|_|___|\n");
         printf("\n************************** TYPING TEST
******************************\n");
         printf("========================== LOG IN
===================================\n");
         printf("(1) Register\n");
         printf("(2) Log In\n");
         printf("(3) Exit\n");
         printf("Enter Choice: ");
         choice = getch() - 48;

         switch(choice)
         {
               case 1:
                     {
                           game.reg();
                           break;
                     }
               case 2:
                     {
                           game.login();
                           break;
                     }
               case 3:
                     {
                           exit(0);
                     }
               default:
                     {
                           printf("\n Invalid input.");
                     }
```

```
            }
    }
    while (choice != '3'); //if the choice is not 3, redisplay the menu
}

// FUNCTION NAME: reg
// CREDITS:
//          Ken Camacho
//
https://www.crazyengineers.com/threads/user-login-and-registration-using-files-in-c.55378
//
// PARAMETERS: None
//
// RETURN VALUE: None
//
// CALLS TO: none
//
// CALLED FROM: main_menu()
//
// METHOD:
//
// This function asks the user to register

void Type :: reg()
{
    FILE *file;
    system("cls");
    printf("\t      _____ _____ _____ _____ _____ _____ _____ _____ \n");
    printf("\t      | __  |   __|   __|           |   __|_   _|   __| __  |\n");
    printf("\t      |           -|   __| |  |-   -|__   | | | |   __|     -|\n");
    printf("\t      |__|__|_____|_____|_____|_____| |_| |_____|__|__|\n");

    printf("\n************************** TYPING TEST
****************************\n");

    //asks user for the necessary information for the registration
    printf("\nEnter First name: ");
    gets(fname);
    printf("\nEnter last name: ");
```

```
    gets(lname);
    username:
    printf("\nEnter Username: ");
    gets(username);

    strcat(username,".txt"); //creates a text file with the username as the filename

    if (file = fopen(username,"r"))//reads the text file and checks if the username inputted already
exist
    {
        printf("\n User Already Exist\n");
        char opt;
        printf("\n Press X to return to main menu or Y to try again: ");
        scanf("%c",&opt);
        if (opt == 'x' || opt == 'X')// if user presses x or X, display main menu again
        {
            main_menu();
        }

        else if(opt == 'y' || opt == 'Y')//if user presses y, ask the user
        {
            goto username;
        }
        system("pause");

        fclose(file);
        reg();
    }

    file = fopen(username,"a+");
    fprintf(file,fname);//displays the first name of the user in the text file
    fprintf(file,"\n");
    fprintf(file,lname);//displays the last name of the user in the text file
    printf("\nEnter password : ");
    gets(password);
    fprintf(file,"\n");
    fprintf(file, password);//displays the password of the user in the text file
    fclose(file);
```

```
    if(!(file = fopen(username,"r")))
    {
            printf("\nUser Not created\n");//if not opened this message will appear
            reg();
    }

    fclose(file);

    printf("\n************************* TYPING TEST
****************************\n");
            system("pause");
}

// FUNCTION NAME: login
// CREDITS:
//          Ken Camacho
//
https://www.crazyengineers.com/threads/user-login-and-registration-using-files-in-c.55378
//
// PARAMETERS: none
//
// RETURN VALUE: none
//
// CALLS TO: none
//
// CALLED FROM: main_menu()
//
// METHOD:
// This function asks the user to login

void Type :: login()
{
    FILE *file;
    char tmp[20];
    system("cls");
    printf("\t      __        _____ _____   _____ _____ \n");
    printf("\t      | | |      |   __| |            |   | |\n");
    printf("\t      | |__|  | | | | |-   -| | | |\n");
```

```
    printf("\t      |_____|_____|_____|  |_____|_|___|\n");
    printf("\n************************* TYPING TEST
****************************\n");
    char user[20], pass[20];
    user:
    printf("\nEnter Username : ");
    gets(user);

    strcat(user,".txt");



    if (!(file = fopen(user,"r")))
    {
            printf("\n User does not exist!\n");//if the username inputted is not a filename of any text
file this message will appear
            char opt;
            printf("\n Press X to return to main menu or Y to try again: ");
            scanf("%c",&opt);
            if (opt == 'x' || opt == 'X')
            {
                    main_menu();
            }

            else if(opt == 'y' || opt == 'Y')
            {
                    goto user;
            }
            system("pause");

            fclose(file);
            login();
    }

    pw:
    printf("\nEnter Password : ");
    gets(pass);

    file = fopen(user,"r");
    while(!feof(file)){
```

```
        fgets(tmp,20,file);
    }

    if (strcmp(pass, tmp) == 0)//for correct password
    {
        printf("\n Log In Successful!\n");
        system("pause");
        usermenu(user);
    }
    else//for incorrect password
    {
        char opt;
        printf("\n Wrong Password\n");
        printf("\n Press X to return to main menu or Y to try again: ");
        scanf("%c", &opt);
        if (opt == 'x' || opt == 'X')
        {
            main_menu();
        }

        else
        {
            goto pw;
        }
        system("pause");
        login();
    }

    printf("\n************************* TYPING TEST
****************************\n");
    system("pause");
}

// FUNCTION NAME: usermenu
// CREDITS:
//
// PARAMETERS:
//   user
//
```

```
// RETURN VALUE: none
//
// CALLS TO: none
//
// CALLED FROM: login()
//
// METHOD:
//        Displays the menu for the game
void Type :: usermenu(char *user)
{
   system("cls");
   fstream f; //iniate file
   f.open(user, ios::app|ios::in);
   int choice;
         do
   {
         system("cls");
         printf("\t        _ _ _ _____ __            _____ _____ _____ _____ \n");
         printf("\t        | | | |  __|  | |    |          |          |  __|\n");
         printf("\t        | | | |  __|  |__|   --| | | | |  __|\n");
         printf("\t        |_____|_____|_____|_____|_____|_|_|_|_____|\n");
         printf("\n\n************************* TYPING TEST
****************************\n");
         printf("============================ MENU
===================================\n");
         printf("(1) Start\n");
         printf("(2) My Profile\n");
         printf("(3) Statistics\n");
         printf("(4) Logout\n");
         printf("(5) Exit\n");
         printf("Enter Choice: ");
         choice = getch() - 48;//key menu variable

         switch(choice)
         {
               case 1:
                     {
                           game.listTest(user);
                           break;
```

```
                }
            case 2:
                {
                    game.myprofile(user);
                    break;
                }
            case 3:
                {
                        game.Statistics(user);
                        break;
                }
            case 4:
                {
                    main_menu();
                    break;
                }
            case 5:
                {
                    exit(0);
                }
            default:
                {
                    printf("\n Wrong Input");
                }
        }
    }
    while (choice != '4');

}

// FUNCTION NAME: myprofile
// CREDITS:
//
// PARAMETERS:
//   user
//
//
// RETURN VALUE: none
//
```

```
// CALLS TO: none
//
// CALLED FROM: usermenu()
//
// METHOD:
//   displays the records of the user
void Type :: myprofile(char *user)
{
   system("cls");
   printf("\t      _____ _____ _____ _____ _____ __ _____ \n");
   printf("\t      |  _  |__  |        |   __|   | | |   __|\n");
   printf("\t      |   __|   -| | |   __|-   -| |__|   __|\n");
   printf("\t      |__|  |__|__|_____|__|  |_____|_____|_____|\n");
   printf("\n\n************************* TYPING TEST
****************************\n");
   ifstream f; //iniate file
   f.open(user);
   f.read((char*)&game, sizeof(game));
   printf("\n Name : %s Last Name: %s", fname,lname);
   printf("\n User name : %s", username);
   char ch;
   f.get(ch);
   printf("\n Your Record \n");
   while (!f.eof()) //displays user's records from file f
   {
          cout<<ch;
          f.get(ch);

   }
   printf("\n Press any key to exit ");
   system("pause");
}

// FUNCTION NAME: listTest
// CREDITS:
//
// PARAMETERS:
//   user
//
```

```
// RETURN VALUE: none
//
// CALLS TO: none
//
// CALLED FROM: usermenu()
//
// METHOD:
//   displays text selection
void Type :: listTest(char *user)
{
   system("cls");
   printf("\t _____ _____ _____ _____ _____          _____ _____ _____ _____ \n");
   printf("\t|   __|_   _|  _  | __  |_   _|  |_   _|   __|   __|_   _|\n");
   printf("\t|__   | | | |    |          -| | |      | | |   __|__   | | |  \n");
   printf("\t|_____| |_| |__|__|__|__| |_|           |_| |_____|_____| |_|  \n");
   printf("\n************************* TYPING TEST
****************************\n");
   fstream f; //iniate file
   f.open("test.txt", ios::in);

   char ch;
   f.get(ch);
   int count = 0;

   start = NULL;
   while (!f.eof())
   {
         Type *newnode = new Type;
         int i = 0;
         while (ch != '\n')
         {
               newnode->tests[i] = ch;
               f.get(ch);
               i++;
         }
         newnode->num = count;
         newnode->next = NULL;

         if (start == NULL)
```

```
        {
                start = newnode;
                last = newnode;
        }
        else
        {
                last->next = newnode;
                last = newnode;
        }
        count++;
        f.get(ch);
    }
    f.close();

    Type *ptr;
    ptr = start;
    printf("\n");
    while(ptr != NULL)
    {
        printf("(%d)",ptr->num);
        printf("%s",ptr->tests);
        ptr = ptr->next;
        printf("\n");
    }
    char choose[2];
    printf("\n Choose Selection ");
    scanf("%s",&choose);
    Test(choose, user);
    usermenu(user);
    system("pause");
    printf("\n************************** TYPING TEST
****************************\n");
}

// FUNCTION NAME: Test
// CREDITS:
//
// PARAMETERS:
//   value
```

```
//   user
//
// RETURN VALUE: none
//
// CALLS TO: none
//
// CALLED FROM: listTest()
//
// METHOD:
//          executes the game and displays result

void Type :: Test(char *value, char* user)
{
    system("cls");
    printf("\n************************** TYPING TEST
****************************\n");
    fstream f,f1;
    int count = 0, wrong = 0, line;
    f.open(value, ios::in);
    if (!f)
    {
            printf("\n Nope");
            return;
    }
    char ch;
    f.get(ch);
    while (!f.eof())
    {
            cout<<ch;
            f.get(ch);
    }
    printf("\n\nHow many lines do you want to write of above selection: ");
    scanf("%d",&line);

    printf("\nStart Writing\n");
    f.close();
    f.open("check.txt",ios::out);
    char text[2000];
    printf("\n");
```

```
system("pause");
clock_t tStart = clock();
while (count != line + 1)
{
	gets(text);
	f<<text;
	f<<"\n";
	count++;
}
clock_t tend = clock();
count = 0;
f.close();


ifstream f10, f20;
string s1,s2, file1, file2, wrongwords;
FILE *wrongfile;
file1 = "check.txt";
file2 = value;
f10.open(file1.c_str());
f20.open(file2.c_str());



char buf[2000];
while (f10>>s2)
{
	f20>>s1;
	if (s1 == s2)
	{
		count++;
	}
	else
	{
		wrong++;

	}
}
printf("\n No of words correct : %d",count);
```

```
    printf("\n No of words Incorrect : %d", wrong);
    float c1 = count;
    float accuracy = c1/(count + wrong) * 100;
    printf("\n Accuracy is : %d percent",accuracy);
    f.close();
    f1.close();



    f.open(user, ios::app);
    f<<"\n";
    f<<"This test result correct words : "<<count<<" Wrong words : "<<wrong<<" Accuracy :
"<<accuracy;
    float time = (tend - tStart)/1000;
    float exact = time/60;
    int speed = count/exact;
    printf("\n Typing Time : %f",time);
    f<<" Test Time : "<<time;
    printf("\n Speed : %d WPM",speed);
    f<<" Speed : "<<speed<<" WPM";
    printf("\n************************** TYPING TEST
****************************\n");
    system("pause");
}

// FUNCTION NAME: Statistics
// CREDITS:
//
// PARAMETERS:
//    user
//
//
// RETURN VALUE: none
//
// CALLS TO: none
//
// CALLED FROM: usermenu()
//
// METHOD:
```

```
void Type :: Statistics(char* user){
    system("cls");
    printf("\t _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ \n");
    printf("\t|   __|_   _|  _  |_   _|          |   __|_   _|       |          |   __|\n");
    printf("\t|__   | | | |   | | | |-   -|__   | | | |-   -|   --|__   |\n");
    printf("\t|_____| |_| |__|__| |_| |_____|_____| |_| |_____|_____|_____|\n");
    printf("\n\n************************* TYPING TEST
****************************\n");

    FILE *stats; //iniate file
    int count=0, index=0, SIZE=100;
    char word[SIZE];
    char c, *ctr;
    printf("\tMISTYPED WORDS\n");
    stats = fopen("wrong.txt", "r");
    c = fgetc(stats);
        while (c != EOF)
        {
        printf ("%c", c);
        c = fgetc(stats);
        }
        fclose(stats);
    printf("\nEnter one of the words above to see how many times it is mistyped: ");
    scanf("%s", &word);


    stats = fopen("wrong.txt", "r");
    while ((fgets(word, SIZE, stats))!= NULL)
    {
        while ((ctr = strstr(word + index, word))!= NULL)
        {
        index = (ctr-word) + 1;
        count++;
        }
    }
    printf("The word %s is mistyped %d times", &word, &count);

    fclose(stats);
    system("pause");
```

}

V. **References**

SyberScribe. (2019). 6 Benefits of Learning How to Type Quickly and Accurately. Retrieved from https://www.syberscribe.com.au/blog/6-benefits-of-learning-how-to-type-quickly-and-accurately/

Harvard. (1969). Harvard Sentences. Retrieved from https://www.cs.columbia.edu/~hgs/audio/harvard.html