#include<iostream>

#include<iomanip>

#include<fstream>

#include<cstdlib>

#define MAX 100

using namespace std;

// Function to read file contents and stores it in respective array

void readFile(int age[], int time[], int& len)

{

// ifstream objects declared to read data from file

ifstream fRead;

// Opens the file for reading

fRead.open("race.txt");

if (!fRead)

{

cout << "\n ERROR: Unable to open the file for reading.";

exit(0);

}

while (!fRead.eof())

{

fRead >> age[len] >> time[len];

// Increase the record counter by one

len++;

}

}

// Function to swap data if invalid

void swapAgeAndTimeInputIfReverse(int age[], int time[], int len)

{

// Loops till number of records

for (int c = 0; c < len; c++)

{

// Checks if current age is greater then the current time

if (age[c] > time[c])

{

// Swapping process

int temp = age[c];

age[c] = time[c];

time[c] = temp;

}// End of if condition

}

}

// Function to return 0 for not qualified

// Returns 1 if age is less than 30 and time taken is less the or equals to 300

// Returns 2 if age is less than 50 and time taken is less the or equals to 360

// Returns 3 if age is more than 50 and time taken is less the or equals to 420

int getRaceQualificationStatus(int age, int time)

{

if (age < 30 && time <= 300)

return 1;

else if (age < 50 && time <= 360)

return 2;

else if (age >= 50 && time <= 420)

return 3;

else

return 0;

}

// Function to return each category qualified counter by reference

void getCount(int age[], int time[], int len, int& notQualified, int& counter30,

int& counter50, int& counterAbove50)

{

// Loops till number of records

for (int c = 0; c < len; c++)

{

// Calls the function get the qualified status

int status = getRaceQualificationStatus(age[c], time[c]);

// If status is 1 then increase the counter30 by one

if (status == 1)

counter30++;

// Otherwise checks if status is 2 then increase the counter50 by one

else if (status == 2)

counter50++;

// Otherwise checks if status is 3 then increase the counterAbove50 by one

else if (status == 3)

counterAbove50++;

// Otherwise not qualified

else

notQualified++;

}

}

// Function to display list

void show(int age[], int time[], int len)

{

// To store the counter for each category

int notQualified = 0, counter30 = 0, counter50 = 0, counterAbove50 = 0;

// Calls the function to get the qualified count

getCount(age, time, len, notQualified, counter30, counter50, counterAbove50);

cout << left << setw(5) << "Age" << setw(6) << "Time" << setw(16) << "Qualified counter" << endl;

// Loops till number of records

for (int c = 0; c < len; c++)

cout << setw(5) << age[c] << setw(6) << time[c] << setw(16) << c << endl;

// Displays the counter

cout << endl << endl;

cout << left << setw(8) << "Amount" << setw(18) << "Category" << endl;

cout << "-----------------------" << endl;

cout << setw(8) << counter30 << setw(18) << "Under 30" << endl;

cout << setw(8) << counter50 << setw(18) << "30 to 50" << endl;

cout << setw(8) << counterAbove50 << setw(18) << "Over 50" << endl;

cout << setw(8) << notQualified << setw(18) << "Didn't Qualify" << endl;

cout << "-----------------------" << endl;

}

int main()

{

// Creates array to store age and time

int age[MAX], time[MAX];

// To store number of records

int len = 0;

// Calls the function to read file contents

readFile(age, time, len);

// Calls the function to swap if invalid age

swapAgeAndTimeInputIfReverse(age, time, len);

// Calls the function to display data

show(age, time, len);

return 0;

}// End of main function

A screenshot of a cell phone

Description automatically generated

A close up of a logo

Description automatically generated