#include<stdio.h>

#include<stdlib.h>

#include<stdint.h>

#include <time.h>

#include <string.h>

#define MAX 20

// create a struct to store elements

struct Node{

    char Player[MAX];

    float Score;

};

//int compare\_name(struct Node\* New\_player,const char \*filename){

//    FILE \*fil = fopen(filename,"r");

//    if(fil == NULL){

//        printf("FAIL: Open fail");

//        exit(1);

//    }

//    char name[MAX];

//    int score;

//    // Rewind file pointer to the beginning

//    rewind(fil);

//  while(fscanf(fil, "%s %f",name,&score) == 2) {

////        printf("Name: %s\n",name);

////        printf("Namemoinhap: %s",New\_player->Player);

//        if(strcmp(New\_player->Player, name) == 0){

//            printf("This name is already in use\n");

//            fclose(fil);

//            return 1;

//        }

//    }

//    printf("Compared \n");

//    fclose(fil);

//    return 0;

//}

struct Node\* Add\_player(){

sua:

    struct Node\* New\_player = (struct Node\*)malloc(sizeof(struct Node));

    if(New\_player == NULL){

        printf("Fail: Memory allocation fail");

        return NULL; // Tr? v? NULL n?u không th? c?p phát b? nh?

    }

    // Nh?p tên ngu?i choi

    printf("Enter Your Player: ");

    scanf("%s",New\_player->Player);

    New\_player->Score = 0;

//    if(compare\_name(New\_player,"Player.txt") == 1){

//        free(New\_player); // N?u tên d? t?n t?i, gi?i phóng b? nh? và th? l?i

//        goto sua;

//    }

    return New\_player;

}

// create the function random number generator

int random(int min, int max){

    return min + rand() % (max + 1 - min);

}

//number separation

void Devide\_number(int test[],int magic\_number){

    // Separate numbers for comparison

    int ngan = magic\_number / 1000;

    int tram = (magic\_number / 100) % 10;

    int chuc = (magic\_number / 10) % 10;

    int donvi = magic\_number % 10;

    // save in array to easy access

    test[0] = ngan;

    test[1] = tram;

    test[2] = chuc;

    test[3] = donvi;

}

// function compare

void compare\_magicNumber(struct Node\* current, int a[]){

    // variable to count the loop

    float dem = 0.00;

    int guest\_number = 0;

    //Create an array to store the number entered by the user

    int check[4] = {0,0,0,0};

    // Enter number

    do{

        printf("Number Guesting is 0 to 9999\n");

        nhaplai:

        printf("Enter number you guest: ");

        if(scanf("%d", &guest\_number) != 1){

            printf("Enter a 4-digit number\n");

            while (getchar() != '\n');

            goto nhaplai;

        }

        int b = guest\_number / 1000;

        Devide\_number(check,guest\_number);

        if(b > 9){

            printf("You enter the number bigger the random number\n");

            printf("Enter again\n");

            goto nhaplai;

        }

        if(guest\_number < 1000 && b != 0 + b){

            printf("Enter a 4-digit number\n");

            goto nhaplai;

        }

        if(guest\_number < 100 && b != 00 + b){

            printf("Enter a 4-digit number\n");

            goto nhaplai;

        }

        if(guest\_number < 10 && b != 000 + b){

            printf("Enter a 4-digit number\n");

            goto nhaplai;

        }

        Devide\_number(check,guest\_number);

        // Check the random number is the same

        for(int j = 0; j < 4; j++){

            if(a[j] == check[j]){

                printf("%d",check[j]);

            }else{

                printf("-");

            }

        }

        printf("\n");

        dem++;

    }while(a[0] != check[0] || a[1] != check[1] || a[2] != check[2] || a[3] != check[3]);

    current->Score = (1/dem)\*100;

}

void savetofile(struct Node\* current){

    FILE \*file = fopen("Player.txt","a+");

    if(file == NULL){

        printf("FAIL: Open fail11111");

        exit(1);

    }

//    while(current != NULL){

        fprintf(file, "%s %.2f \n",current->Player,current->Score);

//        current = current->next;

//    }

    fclose(file);

}

void printtofile(const char \*filename){

    FILE \*file = fopen(filename,"a+");

    if(file == NULL){

        printf("FAIL: Open fail");

        exit(1);

    }

    char name[MAX];

    float score;

    int number\_player = 0 ;

    while (fscanf(file, "%s %f",name,&score) == 2) {

        number\_player++;

    }

    // Rewind file pointer to the beginning

    rewind(file);

    struct Node players[number\_player];

    int i = 0;

    while (fscanf(file, "%s %f",name,&score) == 2) {

        strcpy(players[i].Player, name);

        players[i].Score = score;

        i++;

    }

    for(int i = 0; i < number\_player;i++){

        for(int j = 0; j < number\_player - i;j++){

            if(players[j].Score < players[j+1].Score){

                struct Node tmp = players[j];

                players[j] = players[j+1];

                players[j+1] = tmp;

            }

        }

    }

    if(number\_player < 6){

        for(int i = 0;i<number\_player;i++){

        printf("player: %s, score: %f\n",players[i].Player,players[i].Score);

        }

        printf("There are only %d players\n",number\_player);

    }

    else{

        for(int i = 0;i<5;i++){

        printf("player: %s, score: %f\n",players[i].Player,players[i].Score);

        }

    }

    if (feof(file)) {

        printf("End of file reached.\n");

    } else if (ferror(file)) {

        perror("Error reading from file");

    }

    fclose(file);

}

void manual(){

    printf("----------------------Number Guesting Game------------------------\n");

    printf("Enter\'1\'. Start Game \n");

    printf("Enter\'2\'. Exit Game \n");

    printf("Enter\'3\'. Show High Player \n");

    printf("Enter\'4\'. Delete All Player\n");

    printf("Your selection: ");

}

int main(){

    int choose = 0;

    resel:

//  nhaplai:

    manual();

//    scanf("%d",&choose);

    if(scanf("%d",&choose) != 1){

        printf("Enter 1 or 2 or 3 or 4\n");

        while (getchar() != '\n');

        goto resel;

    }

    switch (choose)

    {

    case 1:

//        printf("Enter\'1\'. Start Game \n");

        // variable to continue the loop

        char con\_ti\_nue;

        do{

            // create the number player must guest

            srand((unsigned int) time(NULL));

            int magic\_number = random (0,9999);

            printf("magic number: %d\n",magic\_number);

            // save in array to easy access

            int a[4] = {0,0,0,0};

            Devide\_number(a,magic\_number);

            // create New\_player

            struct Node\* current = Add\_player();

            // compare

            compare\_magicNumber(current,a);

            printf("%s\n",current->Player);

            printf("%.3f\n",current->Score);

            // ask to continue

            printf("Do you want to play continue? [Y/N]?");

            scanf(" %c",&con\_ti\_nue);

            savetofile(current);

             free(current);

        }while(con\_ti\_nue == 'Y' || con\_ti\_nue == 'y');

        printtofile("Player.txt");

        goto resel;

    case 2:

        exit(0);

        break;

    case 3:

        printtofile("Player.txt");

        goto resel;

    case 4:

    {

        FILE \*f = fopen("Player.txt","w");

        if(f == NULL){

            printf("Erorr: Don't open file");

        }else{

            printf("Delete Success Full");

        }

        goto resel;

    }

        // Detele file

        // int ret;

        // ret = remove("Player.txt");

        // if(ret == 0)

        // {

        //     printf("Delete Successful!!!");

        // }

        // else

        // {

        //     printf("Error: Cannot delete above file");

        // }

        // goto resel;

    default:

        printf("Enter diffrent 1 to 4\n");

        printf("Enter again\n");

        goto resel;

    }

    // printtofile("Player.txt");

    return 0;

}