#include<stdio.h>

// function declarations

int maxDistance(int distance[], int playerNum);

int avgDistance(int distanceP1[], int playerNum1, int distanceP2[], int playerNum2);

//main method

void main()

{

// variables declaration

int games = 10;

int i,j;

int distanceP1[10];

int distanceP2[10];

int playerNum1 = 1;

int playerNum2 = 2;

// fine varaible

FILE\* performanceTest;

// input prompt for loops

for (i = 0;i < 10;i++)

{

for (j = 0;j < 1;j++)

{

printf("Please enter the distance ran by player %d in Game %d\n", playerNum1, i + 1);

scanf("%d", &distanceP1[i]);

printf("Please enter the distance ran by player %d in Game %d\n", playerNum2, i + 1);

scanf("%d", &distanceP2[i]);

}// inner distance input for

}// outer distance input for

// output table headers

printf("Player1 versus Player2\n");

printf("Game\tPlayer1\tPlayer2\t\n");

// output table for loops

for (i = 0;i < 10;i++)

{

printf("%d\t",i+1);

for (j = 0;j < 1;j++)

{

printf("%d\t%d\t", distanceP1[i],distanceP2[i]);

}// inner table display for

printf("\n");

}// outer table display for

// calling functions

maxDistance(distanceP1, playerNum1);

maxDistance(distanceP2, playerNum2);

avgDistance(distanceP1, playerNum1,distanceP2,playerNum2);

// opening file

performanceTest = fopen("performanceTest.txt", "a");

if (performanceTest != NULL)

{

// output table headers to file

fprintf(performanceTest, "Player1 versus Player2\n");

fprintf(performanceTest, "Game\tPlayer1\tPlayer2\t\n");

for (i = 0;i < 10;i++)

{

fprintf(performanceTest,"%d\t", i + 1);

for (j = 0;j < 1;j++)

{

fprintf(performanceTest,"%d\t%d\t", distanceP1[i], distanceP2[i]);

}// inner table display for

fprintf(performanceTest,"\n");

}// outer table display for

fclose(performanceTest);

}

}// main

// functions

// max distance function

int maxDistance(int distance[],int playerNum)

{

// variables

int maxDis,i;

//for loop

for (i = 0; i < 10; i++)

{

if (distance[i] > 0)

{

maxDis = distance[i];

}//if

}//for

return printf("The max distance run by Player %d is %d\n",playerNum, maxDis);

}

// average distance and largest average player function

int avgDistance(int distanceP1[], int playerNum1,int distanceP2[], int playerNum2)

{

// variables

float avgDisP1, avgDisP2;

int i, betterPlayer;

int sumP1 = 0;

int sumP2 = 0;

// for loop

for (i = 0; i < 10; i++)

{

sumP1 = sumP1 + distanceP1[i];

sumP2 = sumP2 + distanceP2[i];

}// for

// average calculation

avgDisP1 = (float)sumP1 / 10;

avgDisP2 = (float)sumP2 / 10;

// player with largest average condition

if (avgDisP1 > avgDisP2)

{

betterPlayer = 1;

}// if

else

betterPlayer = 2;

// return statement

return printf("The average distance ran by Player %d is %.1f and the average distance ran by Player %d is %.0f.\nHence the average distance ran by Player %d is better\n", playerNum1, avgDisP1,playerNum2,avgDisP2,betterPlayer);

}// end of average distance and largest average player