

9. feladat

Legyen adott egy **statisztika.txt** állomány, melynek minden sorában egész értékű térkép koordináták és egy átlagsebesség szerepel

x (szóköz) y (szóköz) v (sortörés)
formában. (Figyelj oda, hogy az utolsó sor után is legyen sortörés!)

Készíts olyan programot, amely beolvassa a koordinátákat és az átlagsebességeket, majd a képernyőre írja, hogy a megadott pontok mentén haladva mekkora távolságot kellett megtenni, hogy az első pontból az utolsó pontba jussunk, és ez mennyi időbe került.

A feladat során feltételezzük, hogy a pontok koordinátái a (0,0) ponttól való **méterben értett x és y** irányú távolságot jelölik. Továbbá feltesszük még azt is, hogy a koordinátákhoz tartozó **v átlagsebesség m/s-ban értendő.**

Segítség:

Két pont távolsága a koordináta rendszerben: $\text{sqrt}((x_1 - x_2)^2 + (y_1 - y_2)^2)$.

A (6,3) és (2,7) pont távolsága: $\text{sqrt}((6 - 2) * (6 - 2) + (3 - 7) * (3 - 7)) = 5.66$ méter.

Az **sqrt()** függvény a **math.h** fejlécállományban található. Linux alatt fordításnál szükséges a **-o** kapcsoló mellett használni kell a **-lm** kapcsolót is. További segítséget ajánlom olvasásra a **man sqrt** paranccsal előhívható leírást terminál/konzol ablakban Linux alatt.

Két pont között a szükséges idő: osztjuk a két pont távolságát (**l**) a második ponthoz beolvasott átlagsebességgel (**v**), $l/v = t$.

Példa:

Ha a fájlban a következő sorok szerepelnek:

```
1 1 0
2 5 3
7 3 2
0 2 6
-1 -7 1
```

Akkor az első pontból a pontok mentén haladva az utolsóig $4.12 + 5.39 + 7.07 + 9.06 = 25.63$ métert tettünk meg. Azaz veszem a távolságokat minden egymás után következő két pontra és összeadom őket. Az összegzett távolság megtételéhez nekünk szükségünk volt $1.37 + 2.69 + 1.18 + 9.06 = 14.30$ másodpercre. Azaz, veszem a távolságokat minden egymás után következő két pontra majd elosztom a második pont átlagsebességével és az így kapott részidőket összegzem.

A képernyőre a kiszámolt adatok az alábbi formában kerüljenek ki:

```
25.63 meter
14.30 masodperc
```

Fontos:

Nagyon fontos, hogy a feladat mellett megtalálható **statisztika.txt** állománnyal ellenőrizd, hogy a programod helyesen működik-e. Ha igen, akkor a fentebb is kiszámolt értékeknek kell kijönnie.

9. feladat

```
/*házi_9*/
```

```
# include<stdio.h>
```

```
# include<stdlib.h>
```

```
# include<fstream.h>
```

```
# include<math.h>
```

```
int main()
```

```
{int i,x[5],y[5],v[5];
```

```
float ido,ut,osszut;
```

```
osszut=0;ido=0;ut=0;
```

```
//Fájl tartalmának beolvasása
```

```
FILE *fp1;
```

```
fp1=fopen("statisztika.txt","rt");i=0;
```

```
while(!feof(fp1))
```

```
{i++;fscanf(fp1,"%d %d %d",&x[i],&y[i],&v[i]);}
```

```
fclose(fp1);
```

```
for (i=1;i<=4;i++)
```

```
{ut=sqrt((x[i+1]-x[i])*(x[i+1]-x[i])+(y[i+1]-y[i])*(y[i+1]-y[i]));
```

```
ido=ido+ut/v[i+1];osszut=osszut+ut;}
```

```
printf("%5.2f meter\n",osszut);
```

```
printf("%5.2f masodperc",ido);
```

```
printf("\n\n");
```

```
system("pause");
```

```
return 0;
```

```
}//main
```

Tömbök deklarálása
(x,y,sebesség)

Fájl tartalmának kiolvasása.

Távolságok meghatározása és összegzése.

Kiiratások.