

Programozás Alapjai 5. próba feladat

1. feladatsor

Szoftverfejlesztés Tanszék

2023, Ősz

Feladat Töltsd le a bíróról a `minta.zip` állományt, majd tömörítsd ki! A `feladat.c` fájlban megtalálod a feladatok megoldás-kezdeményeit. Bővítsd ezt az alább olvasható feladatok alapján! Lehetőség szerint ellenőrizd megoldásod, majd töltsd fel a `feladat.c` fájlt a bíróra!

Kiértékelés A bíró lefordítja a programot, majd lefuttatja azt a feladat pontszámának megfelelő számú tesztessel. Egy tesztet egy bemenet-kimenet pár, amely a megfelelő feladathoz készült. A tesztet akkor helyes, ha az adott bemenethez tartozó kimenet **minden egyes karaktere** megegyezik az előre eltárolt referencia kimenettel. *További feltételek: a program futása nem tarthat tovább 5 másodpercnél, egyszerre nem fogyaszthat többet 16 MiB memóriánál és nem történhet futási hiba (pl. illetéktelen memória hozzáférés).*

Ellenőrzés Feltöltés előtt érdemes ellenőrizni a megoldásod.

1. **Fordítás** Ellenőrizd, hogy a programod lefordul-e! A bíró a `gcc -O2 -static -o feladat feladat.c` paranccsal fordít, érdemes ezt használni. A `-Wall` kapcsoló is hasznos lehet.
2. **Példa tesztesetek** Ellenőrizd, hogy a programod helyesen működik-e! A `minta.zip` tartalmaz a bíró által futtatott tesztesetek közül feladatonként egyet-egyet. Az első feladat teszteléséhez másold a programod mellé az `ex1.be` fájlt `be.txt` néven, futtasd le a programod, majd az így kapott `ki.txt` tartalmát hasonlítsd össze az `ex1.ki` fájlban található referencia kimenettel.
3. **Extra tesztesetek** Ellenőrizd a programod működését további példák segítségével! Néhány további tesztet is elérhető, de ezek csupán ellenőrzésre használhatóak, a bíró nem futtatja őket. Ezek használatához futtasd a programod a `-t` vagy `-test` kapcsolóval, például a `./feladat -test` paranccsal. Csak az első feladat teszteléséhez futtasd a programod a `./feladat -t 1` paranccsal.

1. feladat: KALÓZHAJÓK (4 pont)

Elérkezett az este, egy nagyon kimerítő, kalkulussal teli nap után megérdemlünk egy nyugodt fürdőzést a csodaszép fürdőszobánk fürdőkádjában. Azonban ami ezután történt, arra senki nem számított!

A fürdőkádat megszállták kalózok, akik a fürdőkád hullámozó vizében hirtelen megjelentek gigantikus kalózhajóikkal. Mint minden normális ember, egy ilyen helyzetben elkezdjük megszámlálni a kalózhajókat a fürdőkádunkban. Az alábbi függvény paraméterben várja, hogy hány kalózhajót számoltunk. A feladata, hogy kiírja a standard outputra az alábbi formában: "Kalózhajók száma: <N>", ahol az <N> helyére a kapott érték kerüljön, illetve természetesen a sor végét jelző karaktert se felejtjük le a szöveg végéről.

A számolás nagyon könnyedén ment, ezt a tényt egy 0-s visszaadott értékkel jelezhetjük, ha akarjuk. Ha nem akarjuk, akkor nem kell, csak ebben az esetben nem fogunk rá pontot kapni, úgyhogy én mindenképpen javaslom a jelzését.

Kódold le a függvényt C nyelven! A függvény fejlécén ne változtass! A függvény inputjai a paraméterek.
`int kalozhajok(int);`

2. feladat: KALÓZOK (3 pont)

Amint nézegettük ezeket a hirtelen megjelent kalózhajókat, arra lettünk figyelmesek, hogy a kalózhajókon aprócska kalózocskák szaladgálnak. Kiszemeltük a három legközelebbi hajót, és megszámláltuk a rajtuk lévő kalózokat. Ebből szeretnénk átlagot számítani, hogy komoly tudományos számításokat végezve kiderítsük, hogy megközelítőleg összesen hány kalóz lehet a fürdőkádunkban.

A függvény paraméterben vár 3 értéket, amelyek leírják, hogy a három hajón hány kalózt találtunk.

A függvény feladata, hogy kiírja az outputra a megfigyelésünket az alábbi formában, 2 különböző sorban:

A három legközelebbi hajon levo kalozok: <a>, , <c> db.

Az átlagos legyenység meret: <d>.

A <a>, , <c> helyére kerüljön a kapott 3 paraméter (megfelelő sorrendben), míg a <d> helyére a kapott három érték átlaga, 2 tizedesjegyre kerekítve.

A kiíratás végéről nehogy lemaradjon a sortörés, mert az léket okozhatna a kalózhajókon, ujjujj.

PÉLDA

input: 5 7 4

output:

A három legközelebbi hajón lévő kalozok: 5, 7, 4 db.

Az átlagos legyenység meret: 5.33.

Kódold le a függvényt C nyelven! A függvény fejlécén ne változtass! A függvény inputjai a paraméterek.
`void kalozok(int, int, int);`

3. feladat: HAJÓZÁS (6 pont)

Annyira meglepőek és érdekesek a hirtelen megjelent kalózhajók, hogy gyakorlatilag már mindent el is felejtettünk. A csodálatos kalkulust is, a még csodálatosabb gyakorlatvezetőket, és azt is, hogy amúgy fürdeni akartunk. A figyelmünket megragadta a kád közepés lévő hajó. Na nem azért, mert ez annyira különböző lett volna a többitől, vagy a kalózok furábban viselkedtek volna rajta. Ennél sokkal komolyabb indokunk volt arra, hogy az lett a kiválasztott hajó. Azért, mert az volt a kád közepén.

A kádra tekinthetünk egy 2 dimenziós térképként, ahol az óceánon (azaz a fürdőkádban) a kalózhajók navigálnak egyik célpontról a másikra. És pont ezt figyeltük meg, hogy a kiválasztott hajó milyen útvonalat járt be. A célunk meghatározni, hogy 1 óra alatt mekkora távolságot tett meg a kiválasztott hajó. Kezdetben a (0,0) koordinátáról indul, majd mindig egyenes vonalban halad a következő koordinátára.

A függvény a standard inputról olvassa be a koordinátpárokat egészen addig, amíg (0, 0) párt nem kap meg. Ezt a (0, 0) párt már nem számítjuk bele az útvonalba. A függvény számolja ki és adja vissza az út teljes hosszát, amit a hajó megtett. Minden koordinata egész szám.

PÉLDA

input: 4 0 0 4 2 4 0 0

output: 11.66

magyarázat:

az útvonalnak 3 állomása van: (4, 0), (0, 4) és (2, 4).

1. állomás: (4, 0) → míg ide eljut a hajó a (0, 0) pontból, addig 4 egységnyi távolságot tesz meg

2. állomás: (0, 4) → míg ide eljut a hajó a (4, 0) pontból, addig 5.66 egységnyi távolságot tesz meg

3. állomás: (2, 4) → míg ide eljut a hajó a (0, 4) pontból, addig 2 egységnyi távolságot tesz meg

összesített táv: $4 + 5.66 + 2 = 11.66$.

Kódold le a függvényt C nyelven! A függvény fejlécén ne változtass! A függvény outputja a visszatérési érték.

`float hajozas();`

4. feladat: SZÖRNYEK JAAAAJ (10 pont)

A fürdőkádban lévő kalózoknak ám nem csak a háborgó fürdővízzel kell megküzdeniük, de a veszedelmes vízi szörnyekkel is, amelyek ellepik a fürdőkádat. Ezek a vízi szörnyek igazán hatalmasok, ijesztő tekintetük van és néha még hangokat is adnak ki magukból. Igen, ilyen a legnagyobb ellenségünknek se kívánnánk: gumikacsák. A gumikacsák rettentő nagy hatalommal rendelkeznek, amelyet csak igazán művelt egyének képesek visszaverni, ugyanis a gumikacsák támadásának kivédéséhez másodfokú egyenleteket kell megoldani. És még azt mondják ezt a tudást sehol sem hasznosíthatjuk. Hát mi lenne velünk ha nem értenénk hozzá, és úgy lepnék el a fürdőkádunkat a gumikacsák? Vagy akár éjszaka is meglephetnek álmunkban. Ez veszélyes! A lényeg, hogy nagyon megkedveltük a kalózokat, így szeretnénk őket (illetve magunkat is) megmenteni eme veszedelmes állatoktól.

Az alábbi függvény három paramétert vár, és meg kell oldania az ezen paraméterek által leírt másodfokú egyenletet. A függvény a standard outputra írja ki a két eredményt az alábbi formában: "<x1>;<x2>", természetesen sortöréssel a végén. Az <x1> és <x2> helyére a két megoldás kerüljön (Az <x1> a kisebb).

Amennyiben csak egy megoldás van, úgy az alábbi formában írjon ki: "<x1>". Az eredményeket mindig 2 tizedesjegy pontossággal írjuk ki.

Ha nincs megoldás, akkor "NINCS MEGOLDAS", míg ha végtelen megoldás van, akkor a "VEGTELEN MEGOLDAS VAN" szöveg írjon ki, természetesen a soremeléseket ezúttal se felejtjük el, mert azokkal tudjuk végleg elsüllyeszteni a veszedelmes szörnyeket.

A függvény térjen vissza 0-s értékkel, amennyiben sikerült legalább egy megoldást találni, (-1)-gyel egyébként.

PÉLDA

input: 2, 2, -4

output: -2;1

magyarázat: az egyenlet a $2x^2 + 2x - 4 = 0$, amelynek két megoldása $x_1 = -2$ és $x_2 = 1$.

Kódold le a függvényt C nyelven! A függvény fejlécén ne változtass! A függvény inputjai a paraméterek, outputja a visszatérési érték.

```
int vedelem(float, float, float);
```

5. feladat: ÁGYÚGOLYÓK (7 pont)

A nagy csatában a kalózok rengeteg ágyúgolyót lőttek ki, hogy ezzel is fenn tudják tartani a látszatát, hogy ők győzték le a veszedelmes jószágokat. Persze mindannyian tudjuk, hogy ez nem igaz, de a fürdőszobatükörön üldögélő sirályokat ezzel a trükkel be tudják csapni a kalózok. Pedig természetesen az összes ágyúgolyó egyszerűen visszapattant a kacsákról.

A kalandunk utolsó szakaszában arra vagyunk kíváncsiak, hogy átlagosan hány ágyúgolyót lőttek ki egy hajóról. Ez azért fontos, mert ha elmeséljük a barátainknak, hogy mi történt velünk fürdés közben, akkor esetleg nem fognak nekünk hinni, ha nem mondjuk meg nekik ezen fontos adatokat. Hazudni meg nem szeretnénk, szóval mindenképpen ki kell számolnunk.

A függvény a standard inputról először olvasson be két számot. Az első szám jelenti, hogy hány darab hajó van, míg a második azt jelenti, hogy az eredményt hány tizedesjegy pontossággal kell visszaszolgáltatnunk. A függvény ezután olvasson be annyi számot, ahány hajó van. Ezek a számok jelentik, hogy az egyes hajókról hány ágyúgolyót lőttek ki. Végül a függvény írja ki a standard outputra az átlagot, a megfelelő módon kerekítve, természetesen egy sortöréssel a végén.

Példa input: 3 5 22 12 9

Példa output: 14.33333

Magyarázat:

Az első szám azt jelenti, hogy 3 hajó van

A második szám azt jelenti, hogy 5 tizedesjegy pontossággal kell az eredményt kiírni

A többi a kilőtt ágyúgolyók számát jelenti (22, 12, 9), amelyek átlaga 14.333333...

Kódold le a függvényt C nyelven! A függvény fejlécén ne változtass!

```
void kilott_agyugolyok();
```

6. feladat: IDŐ (5 pont)

A hatalmas csata után a kalózok elhajóztak, a kacsákat elnyelte az áradat (mentek a lefolyóba új célpontok után nézni), így egyedül maradtunk és végre megfürödhettünk. Természetesen az órákig tartó, hatalmas kalandok miatt ez a szokásosnál jóval tovább tartott, egészen pontosan annak duplája volt. Az alábbi függvény feladata megmondani, hogy hány percig tartott a fürdésünk ezen a különleges napon, ha azt tudjuk, hogy általában mennyi ideig szokott tartani. A függvény olvassa be a standard inputról, hogy hány percig szokott átlagosan tartani a fürdés (egész szám), majd írja ki a standard outputra, hogy mennyi ideig tartott most a fürdés (tehát a beolvasott érték duplája). Most lázadjunk egy kicsit, és a kiírás végére ne tegyünk sortörést, menjen a sor a végtelenségig... aztán majd szegény vásárlók napokon keresztül a boltban váraкоznak. És izééé... van itt még valami, amit eddig nem mertem elmondani, és nem is tudom hogyan tudnék hozzákezdeni... Az a helyzet, hogy a nagy kalandok hatására én nagyon megéheztem... Áh, nem, hagyjuk, inkább nem mondom el.

Kódold le a függvényt C nyelven! A függvény fejlécén ne változtass!

```
void dupla();
```