



## Az Informatika-Számítástechnika Tanárok Egyesületének kiadványa

Mottó: „A kompetenciát úgy kell tekinteni, mint olyan általános képességet, amely a tudáson, a tapasztalaton, az értékeken és a diszpozíciókon alapszik, és amelyet egy adott személy tanulás során fejleszt ki magában.”  
(John Coolahan Írország)

### A tartalomból

Előszó .....	1.
Néhány szó a verseny történetéről .....	2.
A 2016/2017. tanévi verseny tapasztalatai .....	3.
Feladatok.....	6.
Regionális forduló (8-10. évf.) ....	6.
Regionális forduló (11-13. évf.)	10.
Döntő .....	14.
Web programozás verseny .....	18.
Mobil programozás verseny .....	20.
Eredmények .....	27.
Regionális forduló.....	27.
Döntő .....	33.
Web-mobil programozás .....	35.

### Előszó

A 2017/2018. tanévben 10. alkalommal rendezzük meg a Dusza Árpád Országos Programozói Emlékversenyt. Ez a kiadvány egy előhírnöke a következő évi, jubileumi évkönyvnek, amit a döntőre tervezünk megjeleníteni.

A 10. évfordulón a döntőre meghívjuk a korábbi évek legeredményesebb versenyzőit és a felkészítő tanáraikat. Ezt megelőzően felkérjük a őket, hogy írjanak visszaemlékezést.

Az évkönyvben közöljük az összes eddigi feladat szövegét, mellékletként megjelentetjük a megoldásokat is. Azt tervezzük, hogy minden feladathoz a 3 legjobb eredményt elérő csapat megoldását közöljük. Szükség esetén kiegészítjük, javítjuk a megoldásokat. Előzetesen egyeztetünk a csapatokkal, kérjük a hozzájárulásukat. Szívesen vennénk, ha ők maguk egészítenék ki a versenyen készített programjukat.

## Néhány szó a verseny történetéről

Az Informatika-Számítás-technika Tanárok Egyesülete 2008-ban szervezte meg először a Dusza Árpád Országos Programozói Emlékversenyt a programozás iránt érdeklődő középiskolások számára.



A névadó, Dusza Árpád tanár úr Miskolcon, a Földes Ferenc Gimnáziumban tanult, majd tanított. Innen ment nyugdíjba, de sajnos nem sokáig élvezhette a nyugdíjas létet. Kollégái, tisztelői már temetésén megfogalmazták, hogy egy versenyt szeretnének indítani az emlékére. Ebből a kezdeményezésből nőtt ki magát ez mostanára már sokak által ismert, igen nívós programozói verseny.

A verseny magas színvonalát a névadó iránti tisztelet és a versenyzők felkészültsége tudja biztosítani. Dusza Árpádról tudni kell, hogy az egyik legeredményesebb versenyfelkészítő informatikatanár volt a 90-es években. 4 éven át minden évben (1995, 1996, 1997 és 1998) volt a Nemzetközi Informatikai Diákolimpián résztvevő tanítványa. Ezek közül 2 évben, 1996-ban két, 1997-ben

három miskolci diák is bekeült a magyar csapatba, ahol nagyon szép eredménnyel végeztek, ezüstérmeket nyertek Fokvárosban.

([http://tehetseg.inf.elte.hu/ioi/ioi\\_main.html](http://tehetseg.inf.elte.hu/ioi/ioi_main.html)).

Közülük hárman, Marhefka István, Tóth László és Újhelyi Gábor rendszeres résztvevői, segítői a programozási versenyünknek az indulástól kezdve.

A tanár úr munkamódszereit ismerők mindannyian azt javasolták, hogy ne egyéni versenyben gondolkodjunk. 2008-ban, amikor először rendeztük meg a versenyt, még nem volt jellemző a csapatverseny, az országos versenyek közül informatikából ez volt az első, amelyen háromfős csapatok indulhattak.

Már az első évben közel 40 csapat jelentkezett, már akkor sikerült elérnünk, hogy az ország legjobbjai nevezzenek a versenyre.

A versenyzőket két korcsoportra bontottuk, de külön kategóriában indulhattak egyes korosztályú csapatok is. Ezt a külön kategóriát később megszüntettük.

Minden évben két fordulót szerveztünk. Az első, regionális forduló 6-10 iskolában zajlott le, a döntő minden évben a miskolci Földes Ferenc Gimnáziumban volt a 10 legjobb csapat részvételével.

2015-ben hirdettük meg először a versenyt mobil programozás és web programozás kategóriában is. Ezekben a versenyeken is háromfős csapatok vehetnek részt, nincsenek korcsoportok, és időben is teljesen elkülönülnek a hagyományos versenytől.

Míg a hagyományos programozói versenyen a megadott helyszínen 4 óra alatt egy komplex feladatot kell megoldania a versenyzőknek mindkét fordulón, a mobil és web versenyen egy még összetettebb feladat megoldására a csapatoknak az általuk választott helyszínen két és fél napjuk van. A döntőben pedig a legjobban sikerült munkák szóbeli bemutatására kerül sor.

A hagyományos programozói versenyen a programozási környezet minden évben megegyezik az emelt szintű érettségien aktuálisan használható környezettel. A web és mobil programozásnál akár a legmodernebb környezet használatára is lehetőség van.

A részletes tudnivalók minden évben a Versenykiírásban és a Versenyszabályzatban szerepelnek, amelyeknek az archivált és az aktuális változata is folyamatosan elérhető az ISZE honlapján, ahogyan a megfelelő fordulók után a feladatok és az eredmények is.

## A 2016/2017. tanévi verseny tapasztalatai

A terveknek megfelelően valósítottuk meg mind a hagyományos versenyt, mind a web és mobil versenyt. Az eddigi legnagyobb létszámú nevezés érkezett be az ország legkülönbözőbb iskoláiból (hagyományos verseny: 71 csapat, web-mobil: 23 csapat). Romániai magyar iskolákból 5 csapat jelentkezett. (Az előző évben Szlovákiából is volt 3 csapat.) A 9-13. évfolyamosok mellett 8. osztályos tanulók is sokan versenyeztek, és nagyon szép eredményeket értek el. Az eddigi legnagyobb számú, 12 helyszínen szerveztük meg a hagyományos verseny regionális fordulóját, a 10 hazai iskola mellett két romániai iskolában. A döntőre hagyományos módon a névadó volt iskolájában, a miskolci Földes Ferenc Gimnáziumban került sor 10 csapat részvételével.

A miskolci Földes Ferenc Gimnáziummal kialakult kapcsolat a szervezésben és a megvalósításban is évek óta gördülékenyen működik. A verseny szakmai előkészítése, a szállás

és étkeztetés lebonyolítása évek óta kialakult rend szerint zajlik problémamentesen.

A web és mobil programozás versenyre is sokan jelentkeztek az idén, de valószínűleg a késői időpont miatt viszonylag sok csapat feladta a küzdelmet, nem küldtek be mun-

kát. Ennek az okait keresve, a jövőben megpróbáljuk időben előbbre hozni ezt a versenyrészt.

Ebben a versenykategóriában a tavalyihoz hasonlóan a regionális forduló két és fél napos volt, a csapatok dönthették el, hogy hogyan és hol dolgoznak (együtt). A döntőt az ISZE oktatótermében tartottuk a legjobb csapatok részvételével. Az előző évihez hasonlóan az idén is profi programozók profi előadásait hallgathatta meg a közönség.

A verseny kezdete óta jellemző, hogy újabb iskolákból jelentkeznek csapatok, jellemző az is, hogy egyes iskolákból egy-két év kihagyás után újra érkezik nevezés. Ez az idén is igaz volt, nem csak a nevezésekre, hanem a döntőbe jutókra is. Ez érdekessé teszi a versenyt, fenn-

marad az érdeklődés tanárban és a diákokban is.

Az első (regionális) fordulóban minden versenyfajtában a csapatoknak közösen kellett egy-egy összetett feladatot megoldaniuk. A hagyományos versenyen egy Markdown-HTML konverter programot kellett írni. A Markdown egy nagyon népszerű, nyílt forráskódú projektekben gyakran használt, jól olvasható jelölőnyelv, amely egyszerű ASCII karakterekkel írja le a weboldalak szerkezetét. A versenyzők által írt programnak az ilyen Markdown-ban megírt „weblapot” kellett átírnia html formára, amit már a böngészők



A döntő résztvevői

meg tudnak jeleníteni. Sok felkészítő tanártól kaptunk olyan visszajelzést, hogy nagyon jónak, életszerűnek tartották a feladatot.



A web programozási verseny első fordulójában egy iskolai tesztkészítő és kiértékelő alkalmazást kellett elkészítenie a csapatoknak. A mobil programozás versenyen pedig egy egészségi állapot felmérő és figyelő alkalmazást kellett megírniuk.

A web- és mobil programozási verseny döntőjén a korábban megírt program bemutatása, prezentálása volt a feladat, a hagyományos versenyen pedig egy olyan programot kellett írni, amelyik egy „okos autóba” szerelt speciális eszköz által küldött helykoordináták és egy térkép alapján meghatározza megadott időközönként az autó sebességét, gyorsulását, megállapítja a sebességtúllépést, és az esetleges szabálytalanságokat.

A mobil programozási feladat megoldásában a versenyzők a saját eszközeiket használták, a web programozási feladatnál egy,

az ISZE-s szervező által biztosított webszerveren kaptak hozzáférést és tárhelyet a csapatok.

A feladatok és javítási útmutatók készítői igen nagy rutinnal rendelkeznek, hiszen a 2008-as kezdés óta végzik ezt a munkát.

A javítótanárok/zsűritagok között folyamatosan van kisebb változás, ami könnyebbé teszi, hogy a feladatok kitűzésében meg tudjunk újulni. Mindig fontos szempont, hogy a gyakorló tanárok mellett bevonjunk gyakorló informatikusokat is. Néhány év kihagyás után az idei évben újra velünk dolgozott a névadó tanár úr egyik tanítványa, Marhefka István, aki társaival 1997-ben és 1998-ban a Nemzetközi Informatikai Diákolimpián Magyarországot képviselte. (1997. Dél-Afrika (Fokváros), (ezüst): Marhefka István Avasi Gimnázium, Miskolc, 1998. Portugália (Setubal) (ezüst): Marhefka István Avasi Gimnázium, Miskolc) Az idei évben kitűzött feladatok ötletei tőle származnak.

A tehetséges tanuló akkor tud fejlődni, ha kellően motivált, ha sokszor meg tudja mérettetni magát. Nagyon jó, hogy informatikából (ezen belül programozásból) több versenyen vehetnek részt a diákok, ez lehetőséget ad arra, hogy az ismereteiket bővítsék. A Dusza Árpád Országos Programozói Emlékverseny nem csak a programozó (abszolút értelemben vett szakmai) kompetenciájukat kéri számon és természetesen mind a felkészülés, mind a verseny során fejleszti, hanem ezen szakmai tudás mellett az életben nagyon fontos egyéb kompetenciákat, (társas kompetencia, idegen nyelvi kommunikáció, matematikai, természettudományi és technológiai, digitális, tanulás tanulása, vállalkozói kompetencia), amelyek elengedhetetlenül fontosak a szakmai



érvényesüléshez. Ezen a versenyen csapatban kell dolgozni, egy nagy összefüggő feladatból egy részt kell megoldani úgy, hogy azt a csapat a feladat végső megoldásához hasznosítani tudja. Részt kell venni a feladat értelmezésében, részfeladatokra bonthatásban, a részfeladat felvállalásában egyaránt. A döntő fordulóban mindemellett prezentálni kell az elkészült munkát, ami nem egyszerű, hiszen itt is együtt kell működni másokkal miközben nagyon komoly szakmai feladatot kell megoldani.

A verseny fenntarthatósága szempontjából nagyon fontos, hogy minél többen tudjanak róla, minél többen kipróbálják. Ennek érdekében igyekszünk minél több alkalommal és minél több formában publikálni a versennyel kapcsolatos információkat.

Az ISZE év eleji tájékoztató levelet az idén is elküldtük az ISZE tagoknak, akik az ország minden táján, több száz középiskolában informatikát tanítanak.

A versenyfelhívást megjelentettük az ISZE honlapján, ahol az eredményeket és a feladatok szövegét is elérhetővé tettük.

Az Inspirációban rendszeresen beszámolunk a versenyről, ahogyan az idén is.

A fentiekén kívül személyes találkozások, ISZE rendezvények alkalmával is bemutatjuk a versenyt.

Évek óta a Nemzeti Tehetségprogram anyagi és szakmai pályázati támogatásával szervezzük a Dusza Árpád Országos Programozói Emlékversenyt. A nevezői létszám

növekedése miatt ezt a támogatást saját forrással is ki kell egészíteni. Az Egyesület próbál szponzorokat felkérni, de a vállalatok többnyire egy-egy informatikai eszközzel támogatják a nyertes versenyzőket. Pénzadományt nem adnak.

Már két évben is az LSI Informatikai Oktatóközpont Alapítvány a web és mobil programozás versenyágban a döntő résztvevőinek adott jelentős összegű jutalmat.

A verseny elismerését igazolja, hogy az ELTE Informatika Kar vezetése felfigyelt rá, és a Média- és Oktatásinformatika Tanszék az idei évben már másodszor tanszékvezetőjével képviseltette magát a döntőn. Dr. Zsákó László tanszékvezető prezentációt is tartott a felkészítő tanároknak és a zsűri tagjainak, amíg a tanulók a versenyfeladatot megoldották. Az eredmények alapján ebben az évben is meghívta az 1. helyezett csapatot az ELTE-n induló tehetségprogramba résztvevőnek, és a Közép-Európai Informatikai Diákolimpia (CEOI) és a Nemzetközi Informatikai Diákolimpia (IOI) válogató versenyére.

Az alábbi linken örömmel látjuk a Dusza Árpád Országos Programozói Emlékverseny döntőjében is szereplő diákokat a legjobbak között.

[http://tehetseg.inf.elte.hu/valogatok/valogatok\\_aktualis.html](http://tehetseg.inf.elte.hu/valogatok/valogatok_aktualis.html)

[http://tehetseg.inf.elte.hu/ioi/ioi\\_main.html](http://tehetseg.inf.elte.hu/ioi/ioi_main.html)

## Feladatsorok

Regionális forduló (hagyományos programozói verseny)

2016. november 19.

9-10. osztályosok feladata

### Feladat

Írjatok Markdown→HTML konvertert!

A markdown egy nagyon népszerű, nyílt forráskódú projektekben gyakran használt, jól olvasható jelölőnyelv, amely egyszerű ASCII karakterekkel jelzi többek között a *dőlt* és **félkövér** szöveget, a címeket vagy hiperhivatkozásokat.

Példa:

*input.md*

```
# Címsor 1

Ez itt egy *kiemelés*. Ez pedig annak a példája,
hogy hogyan lehet **fontos szöveget** írni.

## Címsor 2

Ha szeretnénk, könnyen lehet [linkeket is](http://pelda.hu) írni
markdownban.
```

Ha ezt nyomtatásban tekintenénk meg, akkor a következőt látnánk:

```
Címsor 1
Ez itt egy kiemelés. Ez pedig annak a példája, hogy hogyan lehet fontos szöveget írni.
Címsor 2
Ha szeretnénk, könnyen lehet linkeket is írni markdownban.
```

A fenti markdown esetében a létrehozandó HTML-fájl a következő:

*output.html*

```
<html>
<body>
<h1>Címsor 1</h1>
<p>
  Ez itt egy <em>kiemelés</em>. Ez pedig annak a példája, hogy hogyan lehet <strong>fontos
szöveget</strong> írni.
</p>
<h2>Címsor 2</h2>
<p>
  Ha szeretnénk, könnyen lehet <a href="http://pelda.hu">linkeket is</a> írni markdownban.
</p>
</body>
</html>
```

A konverternek be kell olvasnia az aktuális könyvtárban található *input.md* állományt, el kell végeznie a konverziót, és az eredményt az *output.html* fájlba kell kiírnia.

## Markdown jelölések és azok HTML-megfelelői

A megvalósítandó konverternek nem kell a valóságban előforduló összes markdown jelölést támogatnia. Valójában ez lehetetlen feladat is lenne, hiszen a markdown nem egy szabványos jelölőnyelv, és így sok párhuzamos változata létezik, amelyek különbözőképpen valósítják meg az eredeti koncepciót. Az elkészítendő konverter programnak is egy speciális jelrendszert kell értelmeznie.

A markdown fájl html megfelelőjének minden esetben tartalmaznia kell a szabványos html és body tageket<sup>1</sup> (lásd fenti példa). A tartalomnak a body tagen belül kell elhelyezkednie. A html fájlt tetszőlegesen formázhatjuk, a tageket pl. kezdhethetjük új sorban a könnyebb olvashatóság miatt, de az is lehetséges, hogy a teljes html fájl akár egy hosszú sorba kerül.

A markdown fájlban az egymás utáni sorok egy bekezdést alkotnak, amennyiben nincs közöttük üres sor. Ilyenkor az egyes sorok között tetszőleges whitespace<sup>2</sup> karaktert lehet használni elválasztásra.

Minden markdown fájlhoz érvényes html fájlt kell tudni generálni, ami azt jelenti, hogy a html fájlokban lévő tageknek mindig párban, lezárva kell lenniük, azaz szabályos struktúrát kell alkotniuk. Nem érvényes html pl. a következő fájl, mert nincs lezárva a <p> tag, a <strong> tag és az <em> tag pedig egymást keresztezik:

```
<html>
<body>
  <p>Ez nem <em>ervenyes <strong> xhtml. </em> </strong>.
</body>
</html>
```

A konverternek a következő markdown jelöléseket szükséges támogatnia:

Az elem szerepe, jelentése	Markdown jelölés	HTML megfelelő
Bekezdés	A bekezdések egybefüggő nem üres sorok. Két bekezdést tetszőleges számú üres sor választhat el. Megjegyzés: Üres sor minden olyan sor, amely a whitespace karaktereken kívül más karaktert nem tartalmaz.	<p>bekezdés szövege</p>
Kiemelés	*kiemelendő szöveg* Megjegyzés: Feltételezhetitek, hogy a kiemelés a bekezdésen belül mindig le van zárva.	<em>kiemelendő szöveg</em>
Fontos szöveg	**fontos szöveg**	<strong>fontos szöveg</strong>

<sup>1</sup> A tag a HTML kódban egy egységet jelölő elem. A <html> nyitó és </html> záró tag-ek a html kódot nyitják meg, majd zárják le.

<sup>2</sup> A whitespace karakterek ASCII kódjai: tab (9); kocsivissza (13); soremelés (10); szóköz (32)

	<p>Megjegyzés 1: Feltételezhetitek, hogy a fontos szöveg a bekezdésen belül mindig le van zárva.</p> <p>Megjegyzés 2: Feltételezhetitek, hogy egy szövegrész sem lehet egyszerre kiemelendő szöveg és fontos szöveg is.</p> <p>Megjegyzés 3: Feltételezhetitek, hogy a markdownban nem szerepel ***</p>	
Link	<p>[link szövege](<a href="http://pelda.hu">http://pelda.hu</a>)</p> <p>Megjegyzés 1: Feltételezhetitek, hogy a linken belül a markdownban nem szerepel kiemelés vagy fontos szöveg. Fontos szövegben vagy kiemelésben sem szerepelhet link.</p> <p>Megjegyzés 2: Ha az url rész tartalmaz " karaktert, azt helyettesíteni kell a %22 sztringgel a html-kimenetben.</p> <p>Megjegyzés 3: Feltételezhetitek, hogy a link mindig helyesen szerepel a markdownban, azaz nincs lezáratlan zárójel, azok megfelelő sorrendben követik egymást, és a ] után mindig ( következik whitespace nélkül.</p> <p>Megjegyzés 4: Ha a markdownban szerepel egy [, ], ( vagy ) karakter, az csakis link céljából fordul elő.</p>	<pre>&lt;a href="http://pelda.hu"&gt;link szövege&lt;/a&gt;</pre>
Kisebb jel (<)	<	&lt;
Nagyobb jel (>)	>	&gt;
Címsor 1	Az a bekezdés, amely sor elején a # karakterrel, majd utána egy szóközzel kezdődik.	<pre>&lt;h1&gt; szöveg &lt;/h1&gt;</pre>
Címsor 2	Az a bekezdés, amely a sor elején a ## karakterekkel, majd utána egy szóközzel kezdődik.	<pre>&lt;h2&gt; szöveg &lt;/h2&gt;</pre>
Címsor 3 Címsor 4 Címsor 5 Címsor 6	<p>A Címsor 1 és Címsor 2 mintájára...</p> <p>Megjegyzés: Feltételezhetitek, hogy a # karakter csak a címsor megjelölése céljából és helyes szintaktikával fordulhat elő a szövegben.</p>	A Címsor 1 és Címsor 2 mintájára...



## Beadandó

A program **forráskódja** (a programozási környezettől függően a forráskód több fájl is lehet, esetleg mappák is tartozhatnak hozzá), és a **lefordított fájl verseny** fájlneven, ha a programozási környezet a fordítást támogatja.

A program által használt külső fájlokat mindig az aktuális könyvtárba kell elhelyezni (A program ne tartalmazzon abszolút elérési útvonalat!)

## Kódolási alapelvek

A forráskód minőségét is értékeljük.

Irányelvek, szempontok:

- Egységes kódolási szabályok az azonosítókra:
  - a változók egységes elnevezése (kis- és nagybetűk vagy más speciális karakterek használata),
  - az osztályok egységes elnevezése (objektum-orientált programnyelv esetén),
  - a függvények és eljárások tartalomra utaló elnevezése,
  - a programkód egységes strukturáltsága, tagoltsága.
- A kód minősége (könnyen – emberek számára – érhető illetve karbantartható kód):
  - áttekinthető, lehetőség szerint rövid eljárások, függvények, fájlok,
  - beszédes, tömör elnevezésű azonosítók,
  - objektum-orientált nyelveknél globális változók mellőzése.
- Kommentezés:
  - A kommentezés elsődleges célja, hogy a programban a miért? kérdésre adjon választ. (A mit? kérdésre az azonosítók megfelelő elnevezése és a megfelelően strukturált kód, a hogyan? kérdésre pedig az áttekinthető forráskód ad választ.)
  - A túlzásba vitt kommentezés csökkenti az áttekinthetőséget, a túl kevés komment nehezíti a megértést.
  - Elvárás a változók, osztályok, függvények és eljárások szerepének rövid, értelemszerű kommentezése.

Elérhető pontszám: 100 pont (Ebből a helyes dokumentálás – kommentezés – 15 pont)

Jó munkát kíván a versenybizottság!

## Regionális forduló (hagyományos programozói verseny)

2016. november 19.

## 11-13. osztályosok feladata

## Feladat

Írjatok Markdown → HTML konvertert!

A markdown egy nagyon népszerű, nyílt forráskódú projektekben gyakran használt, jól olvasható jelölőnyelv, amely egyszerű ASCII karakterekkel jelzi többek között a *dőlt* és **félkövér** szöveget, a címekeket vagy hiperhivatkozásokat.

Példa:

*input.md*

```
# Címsor 1

Ez itt egy *kiemelés*. Ez pedig annak a példája,
hogyan lehet **fontos szöveget** írni.

## Címsor 2

Ha szeretnénk, könnyen lehet [linkeket is](http://pelda.hu) írni
markdownban.
```

Ha ezt nyomtatásban tekintenénk meg, akkor a következőt látnánk:

```
Címsor 1
Ez itt egy kiemelés. Ez pedig annak a példája, hogy hogyan lehet fontos szöveget írni.
Címsor 2
Ha szeretnénk, könnyen lehet linkeket is írni markdownban.
```

A fenti markdown esetében a létrehozandó HTML-fájl a következő:

*output.html*

```
<html>
<body>
<h1>Címsor 1</h1>
<p>
  Ez itt egy <em>kiemelés</em>. Ez pedig annak a példája, hogy hogyan lehet <strong>fontos
szöveget</strong> írni.
</p>
<h2>Címsor 2</h2>
<p>
  Ha szeretnénk, könnyen lehet <a href="http://pelda.hu">linkeket is</a> írni markdownban.
</p>
</body>
</html>
```

A konverternek be kell olvasnia az aktuális könyvtárban található *input.md* állományt, el kell végeznie a konverziót, és az eredményt az *output.html* fájlba kell kiírnia.

## Markdown jelölések és azok HTML-megfelelői

A megvalósítandó konverternek nem kell a valóságban előforduló összes markdown jelölést támogatnia. Valójában ez lehetetlen feladat is lenne, hiszen a markdown nem egy szabványos jelölőnyelv, és így sok párhuzamos változata létezik, amelyek különbözőképpen valósítják meg az eredeti koncepciót. Az elkészítendő konverter programnak is egy speciális jelrendszert kell értelmeznie.

A markdown fájl html megfelelőjének minden esetben tartalmaznia kell a szabványos html és body tageket<sup>3</sup> (lásd fenti példa). A tartalomnak a body tagen belül kell elhelyezkednie. A html fájlt tetszőlegesen formázhatjuk, a tageket pl. kezdetjük új sorban a könnyebb olvashatóság miatt, de az is lehetséges, hogy a teljes html fájl akár egy hosszú sorba kerül.

A markdown fájlban az egymás utáni sorok egy bekezdést alkotnak, amennyiben nincs közöttük üres sor. Ilyenkor az egyes sorok között tetszőleges whitespace<sup>4</sup> karaktert lehet használni elválasztásra.

Minden markdown fájlhoz érvényes html fájlt kell tudni generálni, ami azt jelenti, hogy a html fájlokban lévő tageknek mindig párban, lezárva kell lenniük, azaz szabályos struktúrát kell alkotniuk. Nem érvényes html pl. a következő fájl, mert nincs lezárva a <p> tag és a <strong> tag és az <em> tag egymást keresztezik:

```
<html>
<body>
  <p>Ez nem <em>ervenyes <strong> xhtml. </em> </strong>.
</body>
</html>
```

A konverternek a következő markdown jelöléseket szükséges támogatnia:

Az elem szerepe, jelentése	Markdown jelölés	HTML megfelelő
Bekezdés	A bekezdések egybefüggő nem üres sorok. Két bekezdést tetszőleges számú üres sor választhat el. Megjegyzés: Üres sor minden olyan sor, amely a whitespace karaktereken kívül más karaktert nem tartalmaz.	<p>bekezdés szövege</p>
Kiemelés	*kiemelendő szöveg* Megjegyzés: Ha nincs lezárva a * a bekezdés végéig, akkor a bekezdés vége jelenti a * lezárását.	<em>kiemelendő szöveg</em>
Fontos szöveg	**fontos szöveg**	<strong>fontos szöveg</strong>

<sup>3</sup> A tag a HTML kódban egy egységet jelölő elem. A <html> nyitó és </html> záró tag-ek a html kódot nyitják meg, majd zárják le.

<sup>4</sup> A whitespace karakterek ASCII kódjai: tab (9); kocsivissza (13); soremelés (10); szóköz (32)

	<p>Megjegyzés 1: Ha nincs lezárva a ** a bekezdés végéig, akkor a bekezdés vége jelenti a ** lezárását.</p> <p>Megjegyzés 2: Feltételezhetitek, hogy a markdownban nem szerepel ***</p>	
Link	<p>[link szövege](<a href="http://pelda.hu">http://pelda.hu</a>)</p> <p>Megjegyzés 1: Feltételezhetitek, hogy a linken belül a markdownban nem szerepel kiemelés vagy fontos szöveg. Fontos szövegben vagy kiemelésben sem szerepelhet link.</p> <p>Megjegyzés 2: Ha az url rész tartalmaz " karaktert, azt helyettesíteni kell a %22 sztringgel a html-kimenetben.</p> <p>Megjegyzés 3: Feltételezhetitek, hogy a link mindig helyesen szerepel a markdownban, azaz nincs lezáratlan zárójel, azok megfelelő sorrendben követik egymást, és a ] után mindig ( következik whitespace nélkül.</p>	<p>&lt;a href="http://pelda.hu"&gt;link szövege&lt;/a&gt;</p>
Csillag jel (*)	\*	*
Zárójelek: ( ) [ ]	\ \) \ \]	( ) [ ]
Backslash jel (\)	\\	\
Kisebb jel (<)	<	&lt;
Nagyobb jel (>)	>	&gt;
Címsor 1	Az a bekezdés, amely # karakterrel, majd utána egy szóközzel kezdődik. A # karaktert megelőzheti tetszőleges számú whitespace karakter.	<h1> szöveg </h1>
Címsor 2	Az a bekezdés, amely ## karakterekkel, majd utána egy szóközzel kezdődik. A ## karaktert megelőzheti tetszőleges számú whitespace karakter.	<h2> szöveg </h2>
Címsor 3 Címsor 4	A Címsor 1 és Címsor 2 mintájára...	A Címsor 1 és Címsor 2 mintájára...

Címsor 5 Címsor 6		
----------------------	--	--

### Beadandó

A program **forráskódja** (a programozási környezettől függően a forráskód több fájl is lehet, esetleg mappák is tartozhatnak hozzá), és a **lefordított fájl verseny** fájlneven, ha a programozási környezet a fordítást támogatja.

A program által használt külső fájlokat mindig az aktuális könyvtárba kell elhelyezni (A program ne tartalmazzon abszolút elérési útvonalat!)

### Kódolási alapelvek

A forráskód minőségét is értékeljük.

Irányelvek, szempontok:

- Egységes kódolási szabályok az azonosítókra:
  - a változók egységes elnevezése (kis- és nagybetűk vagy más speciális karakterek használata),
  - az osztályok egységes elnevezése (objektum-orientált programnyelv esetén),
  - a függvények és eljárások tartalomra utaló elnevezése,
  - a programkód egységes strukturáltsága, tagoltsága.
- A kód minősége (könnyen – emberek számára – érhető illetve karbantartható kód):
  - áttekinthető, lehetőség szerint rövid eljárások, függvények, fájlok,
  - beszédes, tömör elnevezésű azonosítók,
  - objektum-orientált nyelveknél globális változók mellőzése.
- Kommentezés:
  - A kommentezés elsődleges célja, hogy a programban a miért? kérdésre adjon választ. (A mit? kérdésre az azonosítók megfelelő elnevezése és a megfelelően strukturált kód, a hogyan? kérdésre pedig az áttekinthető forráskód ad választ.)
  - A túlzásba vitt kommentezés csökkenti az áttekinthetőséget, a túl kevés komment nehezíti a megértést.
  - Elvárás a változók, osztályok, függvények és eljárások szerepének rövid, értelemszerű kommentezése.

Elérhető pontszám: 120 pont (Ebből a helyes dokumentálás – kommentezés – 15 pont)

Jó munkát kíván a versenybizottság!

**Döntő (hagyományos programozói verseny)****2017. február 18.****Feladat: Okos autó**

Ma már sok autóba helyezhető olyan speciális eszköz létezik, amely "a gépjármű szabványos diagnosztikai portjára csatlakozik, majd egy felhő alapú informatikai szerverre meghatározott időnként adatokat továbbít. Az eszköz segítségével a jármű képes önmagát diagnosztizálni: érzékeli a sebességváltást, fordulatszámot, a hirtelen irányváltásokat, s ezekből következtetést von le a vezető stílusáról. Emellett, ha lopást észlel, riasztást küld, ha ütközést érzékel, automatikusan segélyhívást indít, a GPS pedig megadja a kocsi pontos koordinátáit. A szenzorokból információ nyerhető ki az autó állapotára vonatkozóan is, így már meghibásodás előtt lehet tudni, mikor szorul szervizelésre a gépjármű. "

(Forrás: [http://www.t-systems.hu/jovokep/201501/merre\\_tartunk](http://www.t-systems.hu/jovokep/201501/merre_tartunk))

A feladatokat egy olyan **alkalmazás** elkészítése, amely egy autó szenzoradatai és egy térkép adatbázis alapján képes az autó állapotának (pozíció, sebesség, gyorsulás) és a vezetői stílusnak (gyorshajtás, hirtelen gyorsítás/fékezés) a megjelenítésére.

**Specifikáció****A térkép**

Az elkészítendő alkalmazás egyik bemenete egy térkép, amelyet – az egyszerűség kedvéért – egy táblázatban ábrázolunk. A táblázat egyes cellái egy-egy útszakaszon a megengedett maximális sebességet mutatják km/h-ban (nemnegatív egész szám). Amennyiben egy érték 0, az úton kívüli területet jelöl.

Az oszlopok és sorok minimális száma 5, maximális száma 20.

A térkép egy cellája egy 10 m x 10 m-es területet jelöl ki. A GPS-koordináták helyett egyszerűsítésként egy, a táblázatra helyezett koordinátarendszert használunk, amelynek az origója a **térkép bal alsó sarka**, az x tengelye vízszintes, az y tengelye függőleges.

**Az utak mindig párhuzamosak a táblázat széleivel, a szélességük 10 m.** Az autó a mozgása során természetesen megváltoztathatja a mozgásirányát, és rákanyarodhat egy másik kijelölt útszakaszra.

Ez a program (alkalmazás) egy új eszköz első teszteléséhez készül, ezért csak kis területen kívánják kipróbálni.

PÉLDA (terkep.txt)

90	90	90	90	90	90	90	90	90	70
50	0	0	0	0	0	0	0	0	70
50	0	0	0	0	0	0	0	0	70
50	0	0	0	0	0	0	0	0	70
50	0	0	0	0	0	0	0	0	70
50	0	0	0	0	0	0	0	0	70
50	0	0	0	0	0	0	0	0	70
50	0	0	0	0	0	0	0	0	70
50	0	0	0	0	0	0	0	0	70
50	50	50	50	50	50	50	50	50	50

A keret a táblázat áttekinthetőségét szolgálja.



## Szenzor adatok

Az alkalmazás másik bemenetét a szenzoradatok szolgáltatják. A szenzor.txt állomány tartalmazza az autó legutolsó útjának adatait. Egy-egy szenzoradat egy sor a fájlban, amely egy-egy eseményt ír le. Háromféle eseményt különböztetünk meg:

- motor indítása: tartalmazza a beindítás időpontját,
- koordináta mintavételezés: tartalmazza az időpontot és a hozzá tartozó aktuális koordinátákat,
- motor leállítása: tartalmazza a leállítás időpontját.

Az időpont mindig az első adat a sorban, **nemnegatív** szám, mértékegysége szekundum. Ezt követi az esemény típusa (MOTOR-BE, KOORD, MOTOR-KI). A KOORD típusú sorokban ezután két **nemnegatív** szám szerepel, ami az autó pozícióját írja le, azaz megadja annak az x és y koordinátáját méterben mérve.

Az egyes sorokban az adatokat szóköz választja el egymástól.

Az autó mindig úton közlekedik. A szenzor koordinátái az érzékelő helyét mutatják, ezért az autót a mozgása során pontszerűnek tekinthetjük.

PÉLDA (szenzor.txt)

0 MOTOR-BE	7.5 KOORD 49 2
0 KOORD 0 2	8 KOORD 56 2
0.5 KOORD 0 2	8.5 KOORD 63 2
1 KOORD 0.25 2	9 KOORD 69.7 2
1.5 KOORD 1 2	9.5 KOORD 75.8 2
2 KOORD 2.25 2	10 KOORD 81.3 2
2.5 KOORD 4 2	10.5 KOORD 86.2 2
3 KOORD 6.25 2	11 KOORD 90.5 2
3.5 KOORD 9 2	11.5 KOORD 94.2 2
4 KOORD 12.25 2	12 KOORD 97.1 2
4.5 KOORD 16 2	12.5 KOORD 99 2
5 KOORD 20.25 2	13 KOORD 99.9 2
5.5 KOORD 25 2	13.5 KOORD 99.99 2
6 KOORD 30.25 2	14 KOORD 99.99 2
6.5 KOORD 36 2	14 MOTOR-KI
7 KOORD 42.25 2	

A példában szereplő autó a 0 s időpontban kerül beindításra. A 0.5 s időpillanatban még a (0;2) koordinátájú pontban tartózkodik, az 1 s időpillanatban a (0.25;2) koordinátájú pontba ér. Ezután folyamatosan halad, először gyorsul, egy nagyon rövid szakaszon állandó sebességgel mozog, majd lassul és megáll. Megállás után, a 14 s időpillanatban a motor is leáll.

Egy út során, amit egy szenzor.txt állomány ír le, többször is lehet motor beindítás és leállítás. Bizonyos autók, ha megállnak (pl. piros lámpánál) automatikusan leállítják a motort, a kuplung megnyomására pedig beindítják.

Az autóba szerelt eszköz bizonyos időközönként mintavételezi a tartózkodási hely koordinátáit. Ha nem változik a pozíció (áll az autó), ugyanaz a koordinátapár egymás után többször is szerepelhet. Előfordulhat, hogy mintavételezések kimaradnak (pl. nincs vétel vagy valamilyen meghibásodás történik.)

„Koordináta mintavételezés” események csak akkor keletkeznek, ha a motor beindított állapotban van.

## Az alkalmazás

Az alkalmazásnak képesnek kell lennie a **térképen** áttekinthetően megmutatni az autó által megtett teljes utat és az autó mindenkori helyzetét. Feltételezhetjük, hogy a szenzor

adatok olyanok, hogy ha az autó útját a térképen ábrázoljuk, akkor bármely két szomszédos mintavételezéshez tartozó térképcella egymással szomszédos vagy egybeesik.

Az **aktuális időpillanathoz** a következő adatokat szükséges megjeleníteni (a térképen vagy mellette):

- a motor állapota (beindítva, leállítva),
- az autó aktuális pozíciója (m-ben),
- az autó aktuális sebessége (km/h-ban) és gyorsulása ( $m/s^2$ -ben),
- az autó sebessége meghaladja-e a megengedett maximális sebességet,
- az autó hirtelen gyorsul vagy lassul-e, azaz a gyorsulása nagyobb-e mint  $2.5 m/s^2$ , vagy kisebb-e mint  $-2.5 m/s^2$

Az autó mozgásadatainak megjelenítéséhez választott időpont a mozgásidőn belül tetszőleges, a felhasználó által kért léptékben növelhető, illetve adott időpillanatra beállítható.

A **teljes útra vonatkozóan** az alkalmazásnak meg kell jelenítenie a következő adatokat:

- a megtett teljes utat (m-ben),
- a kiindulási- és érkezési pont koordinátáit (m-ben),
- az átlagsebességet (km/h-ban),
- a maximális sebességet (km/h-ban),
- az incidensek számát, azaz azt, hogy hányszor fordult elő az út során, hogy az autó vezetője gyorsajtást vagy hirtelen gyorsítást, illetve lassítást követett el. (Ha folyamatosan gyorsan hajt vagy hirtelen gyorsít vagy lassít, az 1db-, ha egyszerre gyorsan is hajt és hirtelen is gyorsít, az 2db incidensnek számít.)

### Sebesség, gyorsulás és pozíció kiszámítása

Az alkalmazásnak egy olyan modellt kell megvalósítania, amely a szenzoradatokban szereplő koordináták alapján képes a mintavételi pontok között is kiszámítani tetszőleges időpontban az autó becsült pozícióját. A modell azzal az egyszerűsítéssel él, hogy **két szomszédos mintavételezés között a gyorsulás állandó**.

A sebesség, a gyorsulás és a megtett út tekintetében az ismert fizikai képletek használhatóak:

$$s = v_0 \cdot t + \frac{a}{2} \cdot t^2$$

$$v = v_0 + a \cdot t$$

ahol  $s$  jelenti a megtett út hosszát,  $v_0$  a kezdősebességet,  $t$  az eltelt időt,  $a$  a gyorsulást,  $v$  pedig a  $t$  időpillanatbeli sebességet.

A modell nem tökéletes, és bizonyos esetekben rendellenességeket produkálhat. Az alkalmazásnak ezekre az esetekre is kell valamilyen megoldást kínálnia.

**Példa:** A fenti szenzor.txt-ben megadott szenzoradatok alapján a sebesség és a gyorsulásértékek a  $t=0.0, 0.5, 1.0, 1.5$  mintavételi pontokban a következők (a sötét háttérű cellák számított értékek):

t	X (m)	Y (m)	v (m/s)	v (km/h)	a ( $m/s^2$ )
0.0	0.00	2.00	0.0	0.0	0.00
0.5	0.00	2.00	0.0	0.0	0.00
1.0	0.25	2.00	1.0	3.60	2.00
1.5	1.00	2.00	2.0	7.20	2.00

A  $t=1.1$  és  $1.49$  közbülső pontokhoz tartozó koordináták, ill. sebesség- és gyorsulás adatok pedig a következők (a sötét háttérű cellák számított értékek):

t	X (m)	Y (m)	v (m/s)	v (km/h)	a (m/s <sup>2</sup> )
1.10	0.36	2.00	1.2	4.32	2.00
1.49	0.98	2.00	1.98	7.13	2.00

Az átlagsebességet a mozgás során megtett útból és az út idejéből számoljuk. Feltételezzük, hogy a motor kikapcsolt állapota nem pihenődött jelent.

#### Beadandó

- A program forráskódja és a lefordított/futtatható állomány (Az állomány neve: `connected_cars`, ami tetszőlegesen rövidíthető)
- A fejlesztői dokumentáció: Szöveges (elektronikus) dokumentum, amelyben ismertetitek a legfontosabb elnevezéseket (változók, osztályok, eljárások, függvények neve), és az egyes részfeladatokban alkalmazott módszereket (az algoritmus működési elvét).

**A bemutatáshoz külön szemléltető anyag készítése (pl. PowerPoint prezentáció) nem kötelező, de ha készül ilyen, azt is be kell adni!**

#### A munka szóbeli bemutatása:

Szemponatok, ajánlott vázlat:

- A feladat előkészítésének bemutatása, a feladatok szétosztásának elvei
- Az elkészített program bemutatása
  - a felhasználó számára
    - A program működésének ismertetése bemutatással
  - a fejlesztő számára
    - A program szerkezetének ismertetése

**Minden csapattagnak részt kell vennie a munka bemutatásában!**

Elérhető pontszám: 150 pont.

**Jó munkát kíván a Versenybizottság!**

## Mobil programozói verseny feladata

2017. március 3. 12:00 - 2017. március 5. 24:00

### Egészséges életmód

Az egészséges állapot fenntartása fontos feladat minden ember számára. A gyorséttermek elterjedésével, a különböző félkész termékek megjelenésével – amelyek lerövidítik a készételhez jutási időt – ez a figyelem szignifikánsan csökken. A boltok polcain elérhető üdítők, édességek, „rágcsák” is csak rontanak a helyzeten. Az oda nem figyelés testsúlygyarapodáshoz, később pedig ebből fakadó betegségekhez vezethet, mint például hipertónia (magas vérnyomás), diabétesz (cukorbetegség), csökkent veseműködés stb. Nagyon sokan a megfelelő folyadékbevitelre sem figyelnek az egészségtelen étkezés mellett, ami kihat a vér hemoglobín-és káros anyag szintjére, valamint csökkent vesefunkciót eredményez.

Sajnos sok embertársunknál már kialakultak az egészségtelen táplálkozás következményei, számukra legfontosabb az állapotuk javítása, vagy romlás megelőzése, lassítása.

### Feladat

A MINDENT AZ EGÉSZSÉGÉRT mikrovállalat olyan mobil alkalmazás elkészítésével bízta meg a fejlesztőit (Benneteket), amellyel nyomon lehet követni egy ember naponkénti táplálkozását, folyadékfogyasztását és vérnyomását.

### Specifikáció

- A programban lehessen rögzíteni az elfogyasztott ételeket, azok adagjait. A rögzítéskor kerüljön be a fogyasztás időpontja is, ami lehet automatikus (pl.: most került elfogyasztásra), vagy a program kérheti a felhasználótól azt.
- A programban lehessen rögzíteni az elfogyasztott italokat, azok mennyiségét. A rögzítéskor kerüljön be a fogyasztás időpontja is, ami lehet automatikus (pl.: most került elfogyasztásra), vagy a program kérheti a felhasználótól azt.
- A programban lehessen rögzíteni a vérnyomást is, annak időpontjával együtt, ami lehet automatikus (pl.: most történt a mérés), vagy a program kérheti a felhasználótól azt.
- A programban lehessen rögzíteni a testsúlyt is naponta.
- A program napi, heti, havi lebontásban, valamilyen szemléltetési módon prezentálja a fenti adatokat, nem feltétlenül egy képernyőn. A statisztika készítésekor vegye figyelembe az illető nemét. A nemekhez általánosan szükséges napi kalória mennyiséget is megállapítottak, amit vegyen figyelembe a program, és jelezze, ha aznap túllépte az alapvető energiaszükségletet.
- A programban lehessen rögzíteni esetlegesen plusz tevékenységeket is azok energiaszükségletével együtt, amit a napi fogyasztott kcal mennyiségbe be kell számítani (le kell vonni).
- A program délután 4 órakor jelezzen valamilyen módon a felhasználónak, ha aznap még nem rögzített adatokat, de ha nem kerül be aznap adat, az a működést ne befolyásolja.
- A program a rögzítések után jelezhet, ha már az aznapi bevitt energiamennyiség túllépte a napi szükségletet, de hiány esetén is értesíthet. Például délután 4 órakor még csak 8 deciliter folyadékbevitelnél tart.
- A program figyelmeztessen, ha magas vérnyomást észlel. Ezt megteheti kiugróan magas értékénél, de ha egy hónapban háromszor kerül ilyen adat rögzítésre, akkor is jelezzen, mert ez a kezdődő magas vérnyomásos betegség jele lehet.

### Általános információk

- Általános tézis, hogy egy átlagos nő energiaszükséglete 2000 kcal, míg egy férfié 2500 kcal. De ez lehet alkat, kor, végzett munka alapján is kalkulálni, de ez nem feltétlenül elvárás.
- Magas vérnyomásnak a 140/90 Hgmm-nél magasabb értéket tekintjük. Ez lehet 150/85 Hgmm, de lehet 120/110 Hgmm, vagy 160/120 Hgmm is.
- Az ételeket célszerű előre rögzíteni a hozzájuk tartozó energiatartalommal kcal egységben, a bevételkor ebből válasszon a felhasználó, és csak az adagot/mennyiséget adja meg.
- Egy ember napi folyadékigénye 2,5-3 liter.

### Egyéni ötlet

A szoftverhez bármilyen egyedi ötlet hozzáadható, ami feladathoz kapcsolódik, ez pontozásban külön meg is jelenik.

### Értékelési szempontok

- A feladat kidolgozása, megoldása (60 %)
- Saját ötlet megvalósítása (15 %)
- Az alkalmazás esztétikus kialakítása (20 %)
- Kommentezés, a kód minősége, olvashatósága, felhasználói dokumentáció (5 %)

## A Web programozói verseny feladata

2017. március 3. 12<sup>00</sup> - 2017. március 5. 24.00

### Tesztek kitöltése weblapon

A diákok tudásának felmérésére már nagyon régóta használunk az oktatásban tesztek.

A tesztek kitöltésének módja a papíron történő jelölés helyett a számítógépen való "tippelés" is lehet. Ennek a módszernek sok előnye van.

A tanár számára például az, hogy a teszt kiértékelését is a számítógép végezheti el. Az eredményeket személyre lebontva vagy akár csoport szintű összesítésben is elő tudja állítani egy jól megírt program. A diák számára tanulságos, hogy a kitöltés után azonnal visszajelzést kaphat az eredményéről, így önállóan is gyakorolhat. Ez segíthet a felkészülésben, a tananyag mélyebb megértésében, a vizsgákra történő felkészülésben. A tesztek kitöltésére akár a tabletje vagy telefonja is alkalmas lehet.

Egy weblapon kitölthető teszt összeállítása azonban összetett programozási feladat.

A tanári munkát nagyban segítené egy olyan környezet, amely egy szövegfájlból képes egy webes teszt-oldalt automatikusan előállítani, vagy egy ilyen fájl előállítását interaktív módon segíti.

### Csapatotok egy ilyen program elkészítésére kaptok megbízást.

A program többféle tesztípust kell, hogy kezeljen, a tesztek elkészítő tanárok és a tesztet kitöltő diákok szempontjait, igényeit is figyelembe kell vennie. Elvárás az is, hogy a képernyőn megjelenő szöveg jól olvasható, megfelelően tagolt és esztétikus legyen.

A tesztben elhelyezhető kérdések típusai:

1. **Egyszerű választás:** A kérdés után felkínált lehetőségek közül egyetlen válasz a helyes, ezt kell kiválasztani "rádiógomb" bejelölésével. A jó válasz 1 pontot ér.
2. **Többszörös választás:** A kérdés után felsorolt válaszok közül több is helyes lehet, ezeket kell "jelölőnégyzet" segítségével kiválasztani. A kérdés annyi pontot ér, ahány jó választ kell megjelölni. A téves jelölések -1 pontot érnek. Negatív összpontszámot nem adunk, ha ennyi jönne ki, akkor nulla pontot kell adni. Ha minden lehetőséget bejelölt, akkor is 0 pont jár - kivéve azt az esetet, ha ez a megoldás.
3. **Mondat kiegészítése legördülő menüben megjelenő szavakkal:** Egy mondatba az odaillő szót kell kiválasztani egy legördülő menüből. Itt csak egy jó válasz lehet. A kérdés több mondatból is állhat, egy mondatban több helyen is lehet kiválasztandó szó. A kérdésre annyi pont jár, ahány legördülő menüből választhatott a diák.

### A tesztek elkészítésének módja

#### 1. Speciális szerkezetű szövegfájl összeállítása és feltöltése:

A tesztet leíró szövegfájl nagyon egyszerű szerkezetű, csak néhány szabályt kell megjegyezniük az összeállítást végző tanároknak.

#### Ezek a szabályok a következők:

A teszt leírását az első 4 sor tartalmazza.

- Az első sorban a tantárgy neve áll, amelyhez a teszt leginkább illeszkedik.
- A második sorban a témakör megnevezése szerepel. Ez szabadon megadható rövid szöveg, pl: Kombinatorika, Szerves vegyületek, A számítástechnika története
- A harmadik sor a teszt nehézségi szintje, ez ALAP, KÖZÉP EMELT stb. lehet.
- A negyedik sorban a tesztet elkészítő tanár neve szerepel.

Ezt követik a kérdések.

Minden teszt minimum 2 kérdést tartalmazzon!

Egy kérdés elejét a sor elején elhelyezett :KX: , a végét a sor elején elhelyezett :V: jelezze. Az X a kérdés típusára utal, ami 1, 2 vagy 3 lehet.



( 1-egyszerű választás, 2-többszörös választás, 3-mondat(ok) kiegészítése menüből.)

A következő rész a kérdés szövege, amely többsoros is lehet. A képernyőn a szövegfájlban alkalmazott sortördelés jelenjen meg!

Az 1-es és a 2-es típus esetén a kérdés után a válaszok szerepelnek 1-1 sorban. A sor elején - (mínusz) jel legyen a hibás válaszoknál, + (plusz) legyen a helyes válasz(ok)nál! Egyszerű választás esetén természetesen csak egy + jellel kezdődő sor lehet a válaszok között.

A 3-as típusú teszt esetén a mondatban a kívánt helyen/helyeken [ ] jelek között kell elhelyezni a lehetséges válaszokat, amelyek a legördülő menüben majd megjelennek. A válaszokat # (kettős kereszt) választja el egymástól. A helyes válasz előtt + (plusz) jel szerepeljen! A válaszok között, továbbá a [ , ], + és # jel előtt és után szóközök is lehetnek.

A szövegfájlban a teszt megformázására, kiemelésére néhány HTML kódot engedélyezzen a rendszer! Ezek a *dőlt* <i> </i>, a **félkövér** <b> </b>, az aláhúzott <u> </u> szövegek és a vízszintes vonal <hr> lehetnek. Minden más < > jelek között elhelyezett dolgot egyszerűen hagyjon figyelmen kívül a program a tesztlap megjelenítésekor!

A HTML formázás a teszt készítőjének lehetősége és a felelőssége. Ha például nem zár le egy <b>-vel kezdődő részt, akkor ezt nem kell a programnak hibaként kezelnie. Arra azért ügyeljen a program, hogy egy lezáratlan HTML tag a következő kérdés megjelenésére már ne legyen hatással!

## 2. Weblapon történő interaktív összeállítás

Ha valaki a speciális szövegfájl nem tudja elkészíteni, akkor kapjon lehetőséget arra, hogy weblapon interaktívan is összeállíthasson egy több kérdésből álló tesztet. Ebben az esetben elegendő csak az 1. és a 2. kérdéstípust megvalósítani.

A weblapon megadható kérdések és válaszlehetőségek alapján a program a fenti leírásnak megfelelő szerkezetű szövegfájl kell, hogy előállítson. A szövegfájl legyen letölthető!

## A tesztek kitöltése

A teszt megírásának szabályozása: (a kitöltés módja)

1. Egyszerre csak a teszt egyetlen kérdése látszik a képernyőn. A válasz megadása után a kérdés és a beírt válasz már nem látható, visszalépni, módosítani a válaszon már nem lehet. A kitöltés során megjelenik, hogy a teszt hányadik kérdésénél járunk.
2. A teszt valamennyi kérdése látszik a képernyőn (scrollozni lehet), az összes választ egy gomb lenyomása után küldi el a szervernek.

## Az eredmények megtekintése: (visszajelzés mértéke)

1. A válaszok elküldése után semmiféle visszajelzés nincs az eredményről, csupán az adatok beérkezését nyugtázzuk.
2. Az összes válasz beérkezése után egy százalékos eredményt ír csak ki valamennyi kérdés találatait összegezve.
3. Minden kérdésre megadja a találati százalékot és a kérdéssor összesített eredményét is. A kérdéseket emlékeztetőül megjeleníti.
4. Minden kérdésre megadja a diák választ, és ha az helytelen vagy nem teljes, akkor a helyes eredményt is közli. A kérdéseket emlékeztetőül megjeleníti.

## Teszt elérhetővé tétele

A feltöltött szövegfájlok alapján készülő tesztek nem lesznek elérhetőek a diákok számára automatikusan. Egy tanár a saját tesztjeit teheti elérhetővé egy tetszőleges cím megadásával. Ezt a címet látják majd a diákok, amikor a kitölthető tesztek között válogatnak. A tanár beállíthatja a kitöltés módját (2-

féle lehetőség) és a visszajelzés mértékét (4-féle lehetőség). Legyen lehetősége a teszt színvilágának kiválasztására 3-4 sablon közül!

Egy tesztet több formában is elérhetővé lehessen tenni!

Legyen mód egy kiadott elérhetőség visszavonására!

A teszt kitöltésének időtartamát is figyelje a program! Legyen lehetőség a kitöltés idejének korlátozására, ezt percekben lehessen megadni! Ebben az esetben a teszt kitöltése során a hátralévő idő is jelenjen meg a képernyőn, vagy legyen mód ennek a lekérdezésére!

### Szerepek és azok feladatai, lehetőségei

#### Tanár:

A tanárok a bejelentkezéshez egyedi azonosítókat kapnak. Az egyszerűség kedvéért most ezek legyenek: tanar1,tanar2,...tanar10. A tanárok bejelentkezés után egy nekik elkészített lapról az alábbi lehetőségek közül választhassanak:

- Egy tesztet tartalmazó szövegfájl feltöltése
- Egy több kérdésből álló teszt összeállítása interaktívan weblapon
- Egy korábban feltöltött vagy összeállított teszt letöltése szövegfájlként
- Saját teszt elérhetővé tétele és a kitöltés feltételeinek beállítása
- Egy elérhetővé tett teszt visszavonása
- Egy diák egy tesztjének részletes megtekintése (a pontszámok kérdésenként, a diák válaszai, a kitöltés időtartama, az összpontszám és a százalékos eredmény, hányszor töltötte ki ezt a tesztet, és ez hányadik verziója)
- Egy teszt eredményeinek összesített megjelenítése (hányan töltötték ki, mennyi az átlagos összpontszám és százalék)
- Egy teszt eredményeinek kérdésenkénti összesítése
- Részletes segítő oldal lekérése a tesztek elkészítéséhez, amely tartalmazza a feltöltendő szövegfájl szerkezetének leírását, és kiter a weblapon való teszt elkészítésének menetére.

#### Diák:

A diákok a bejelentkezéshez egyedi azonosítót kapnak. Az egyszerűség kedvéért most ezek legyenek: diak1, diak2, ...diak100

- Válogatás a felkínált tesztek között (tantárgyanként, nehézségi szintenként)
- Teszt kitöltése (a teszthez beállított feltételekkel)
- Saját eredményeinek megtekintése egy teszt kitöltése után (ha a teszt beállítása ezt engedi)
- Lista lekérése a korábban kitöltött tesztjeiről és azok eredményéről

#### Rendszergazda:

A rendszergazda a bejelentkezéshez egyedi azonosítóval rendelkezik. Az egyszerűség kedvéért ez most legyen: admin

- Áttekintés a tanárok tevékenységéről (hány tesztjük van, ezek közül mennyi van most kitölthető státuszra beállítva, hányszor töltötték ki ezeket)
- Áttekintés a diákok tevékenységéről (melyik tesztet töltötték ki és hány alkalommal)
- Áttekintés a be nem fejezett és az időn túli tesztekéről (melyik diák melyik teszt)
- Áttekintés a szerveren lévő tesztfájlokról darabszámáról tantárgyanként, tanáronként, témánként

### Hibák kezelése

A feltöltött szövegfájlban lehetnek formai hibák. Ezeket vegye észre a program és jelezze a tanárnak, a feltöltött hibás fájlt pedig tekintse semmisnek.

#### A hibák lehetnek pl.:

- kérdés elejének vagy végének hiánya ( :KX: :V: )
- a kérdés típusa nem megfelelő (csak 1 vagy 2 vagy 3 lehet)

- nincs megadva, melyik a jó válasz
- több jó válasz is meg van adva 1-es és 3-as típusú kérdés esetén
- csak 1 válaszlehetőség van megadva

#### **A weboldalak megjelenése**

A program használata során a képernyőn mindig legyen látható, hogy milyen jogosultsággal/azonosítóval léptünk be, melyik tesztet töltjük ki éppen, (tantárgy, témakör, szint, tanár neve), mennyi a pontos idő, mennyi időnk maradt.

#### **A program elkészítése**

Törekedni kell a jól olvasható programkód megírására, a beszédes változónevek használatára! A kód értelmezéséhez kommenteket is el kell helyezni a forrásban!

A feladat leírásában nem részletezett problémás esetek kezelését a programozói csapatnak kell megoldania, és ennek módját a forrásban dokumentálja is! (pl. azonos nevű fájlokat próbálnak meg feltölteni, a teszt kitöltése megszakad, stb.)

**MINTA**

Egy tesztet tartalmazó szövegfájl pl. ilyen lehet:

---

Matematika

Összeadás, kivonás

ALAP

Tanár1

:K1:

Mennyi  $11+5$  ?

Jelöld be a helyes értéket!

- 15

+ 16

- -16

-18

:V:

Üres sorok is lehetnek két kérdés között.

Megjegyzés is lehet itt.

:K2:

Kivontunk 10-ből egy <b>tetszőleges</b> számot.

Az eredményre vonatkozó állítások közül jelöld be a helyeseket!

(Több jó válasz is lehet!)

- az eredmény biztosan negatív

+ az eredmény lehet pozitív

- az eredmény biztosan kisebb 10-nél

+ az eredmény lehet 10-nél kisebb

+ az eredmény lehet 10-nél nagyobb

:V:

Megjegyzés lehet itt is.

:K3:

Az alábbi mondatokhoz a listából válaszd ki a megfelelő kifejezést!

Ha páros számhoz adunk párosat, akkor biztosan [páratlan #+páros] számot kapunk.

Ha páratlan számból vonunk ki egy [ + páratlan #páros] számot, akkor <font color="red">páros számot kapunk.

:V:

Itt a vége a fájlnek

---

A mintafájl magyarázatokkal:

A szövegfájl tartalma	Magyarázat
Matematika	1. sor: tantárgy megnevezése
Összeadás, kivonás	2. sor: téma
ALAP	3. sor: szint
Tanár1	4. sor: tanár neve
:K1:	Kérdés kezdete, a típusa 1, tehát egyszerű választás.
Mennyi $11+5$ ?	A kérdés szövege, a sortörés helye számít!
Jelöld be a helyes értéket!	
- 15	Hibás válasz.
+ 16	A jó válasz. A sor elején lehetnek szóközök.
- -16	Hibás válasz. A sor elején lehetnek szóközök.
-18	Hibás válasz.
:V:	Kérdés vége Ez a kérdés 1 pontot ér.
	Itt bármi lehet, az nem kerül ki a weblapra - üres sorok, megjegyzések stb
Megjegyzés is lehet itt.	
:K2:	A 2. kérdés kezdete, a típusa 2 tehát többszörös választás.
Kivontunk 10-ből egy <b>tetszőleges</b> számot.	A <b>“tetszőleges”</b> szó félkövér lesz a tesztlapon.
Az eredményre vonatkozó állítások közül jelöld be a helyeseket!	Új sorban jelenik meg.
(Több jó válasz is lehet!)	Új sorban jelenik meg.
- az eredmény biztosan negatív	
+ az eredmény lehet pozitív	

- az eredmény biztosan kisebb 10-nél	
+ az eredmény lehet 10-nél kisebb	
+ az eredmény lehet 10-nél nagyobb	
:V:	Kérdés vége. Ez a kérdés 3 pontot ér.
Megjegyzés lehet itt is.	
:K3:	A 3. kérdés kezdete. A típusa 3, tehát kiegészítendő mondat(ok).
Az alábbi mondatokhoz a listából válaszd ki a megfelelő kifejezést!	
	Ez az üres sor már a kérdésen belül van, ez a tesztlapon is megjelenik.
Ha páros számhoz adunk párosat, akkor biztosan [páratlan #+páros] számot kapunk.	
Ha páratlan számból vonunk ki egy [ + páratlan #páros] számot, akkor <font color="red">páros számot kapunk.	A <font color="red">nem jelenik meg a weblapon és nem veszi figyelembe a szöveg színének beállításához a program, mert nem megengedett HTML tag. A +, a #, a [ és a ] jelek előtt és mögött lévő szóközt sem tekintjük hibának.
:V:	Ez a kérdés 2 pontot ér.
Itt a fájl vége	A tesztlapon ez már nem jelenik meg.



## Eredmények

### Regionális forduló (hagyományos verseny)

#### 8-10. évfolyam

	A csapat jel-igéje	A regionális fordulóban elért pontszám	Iskola	Az iskola székhelye	Csapattagok		Felkészítő tanárok
I.	BACKSLASCH	95	Nyugat-magyarországi Egyetem Bolyai János Gyakorlói Általános Iskola és Gimnázium	Szombathely	Péteri Bence	10.	Dobre Norbert
					Poór Máté	9.	
					Dobos Dominik	9.	
II.	Bitmozgatók	90	Esztergomi Dobó Katalin Gimnázium	Esztergom	Szén Máté Gábor	9.	Szurdi Gábor
					Zrupkó Zoltán	9.	
					Szerencsés Dániel	9.	
III.	nosleepnolife	87	Kőbányai Szent László Gimnázium	Budapest	Rádai Ronald	9.	Nagy Tibor Imre
					Szedlák Bence Patrik	9.	
					Vörös Asztrik	9.	
IV.	Hókuszpókusz	85	Szent László Katolikus Gimnázium, Szakgimnázium, Általános Iskola, Kollégium és Óvoda	Kisvárd	Molnár Máté	8.	Dajka Miklós
					Jedla Martin	8.	
					Kovács Marcell	9.	
V.	SZIGet	82	Szent István Gimnázium	Budapest	Baczó Domonkos	9.	Borbásné Penke Judit Szalayné Tahy Zsuzsa
					Klapper Zsombor	9.	
					Bui Quanganh Krisztián	9.	
VI.	pyjva	80	Szolnoki Műszaki Szakképzési Centrum Pálffy-Vízügyi Szakgimnáziuma	Szolnok	Takács Sándor Attila	10.	Englert Ervin
					Kosztá Róbert	10.	
					Dénes Petra	10.	
7.	Korbenisztan	73	Lovassy László Gimnázium	Veszprém	Bucsi Benjamin	10.	Báder Anikó
					Piller Trisztán	10.	
					Takács Kornél	10.	
8.	endlessloop	57	Városmajori Gimnázium és Kós Károly Általános Iskola	Budapest	Lévai Zalán Bálint	10.	Bärnkopf Bence
					Bertók Attila	10.	
					Balogh Koppány	10.	
9-10.	cipősdoboz	50	Városmajori Gimnázium és Kós Károly Általános Iskola	Budapest	Rakolcza Péter	10.	Bärnkopf Bence
					Sáfrán Ábel	10.	
					Kovács Domonkos	10.	
9-10.	PePóTa	50	Táncsics Mihály Gimnázium	Kaposvár	Pete Dávid	10.	Biczóné Lengyel Beáta Raskoványi Miklós
					Pósa Péter	10.	
					Tánczos Gergely	10.	
11.	Tunyi	46	Kőbányai Szent László Gimnázium	Budapest	Csepán Botond	10.	Nagy Tibor Imre
					Deák Árpád	10.	
					Nógrádi Viktor	10.	

Döntőbe jutottak

## Dusza Árpád Országos Programozói Emlékverseny

12.	Gőzmacska	42	Kecskeméti Szakképzési Centrum Kándó Kálmán Szakgimnáziuma és Szakközépiskolája	Kecskemét	Tóth Ramóna	10.	Trepák Ildikó
					Tóth István	10.	
					Polyák Szabolcs	10.	
13-14.	FaTó	40	Táncsics Mihály Gimnázium	Kaposvár	Fazekas András	10.	Biczóné Lengyel Beáta Raskoványi Miklós
					Tóth Tóbiás	10.	
13-14.	Igény	40	Kőbányai Szent László Gimnázium	Budapest	Ambrus-Dobai Márton	10.	Nagy Tibor Imre
					Berta Gábor Pál	10.	
					Orbán András Kende	10.	
15.	Búszted Animálsz	34	Budapesti Műszaki Szakképzési Centrum Neumann János Számítástechnikai Szakgimnáziuma	Budapest	Szécsey Péter	10.	Kovácsné Murányi Katalin Schmidt Vilmos
					Kollár Ádám	10.	
					Simon Patrik	10.	
16.	Kecsketej pudding	32	Debreceni Szakképzési Centrum Mechwart András Gépipari és Informatikai Szakgimnáziuma	Debrecen	Dremák Gergely	10.	Péter Miklós Nagy Keve
					Jánószky Máté Levente	10.	
					Székely Csaba	10.	
17-20.	Káldeusok	25	Kőbányai Szent László Gimnázium	Budapest	Bodó József	10.	Nagy Tibor Imre
					Birtalan Kinga	10.	
					Kiss Gergely Domonkos	10.	
17-20.	Cicamix	25	Kőbányai Szent László Gimnázium	Budapest	Bézi Ákos	9.	Nagy Tibor Imre
					Vargha Csongor Csaba	9.	
					Neszlényi Kálmán Balázs	9.	
17-20.	010	25	Kempelen Farkas Gimnázium	Budapest	Tersztenyák Balázs	9.	Olejnik Árpád
					Kolozsvári Márton	9.	
					Papp Márton	9.	
17-20.	trumpteam	25	Lovassy László Gimnázium	Veszprém	Borsos Zsófia	9.	BáderAnikó Szabó László
					Jeges Gábor	9.	
					Simon Noel	9.	
21.	Ellazuljunk	20	Lovassy László Gimnázium	Veszprém	Birkás Balázs	9.	Szabó László
					Horváth András	9.	
					Hunyadi Zsombor	9.	
22.	Isack	18	Lovassy László Gimnázium	Veszprém	Márkus Bánk	9.	Báder Anikó
					Sebők Bálint	9.	
					Troják Bálint	9.	
23-25.	Komolytalan alakzatok	15	Szent István Gimnázium	Budapest	Hadházy Áron Tamás	9.	Borbásné Penke Judit Szalayné Tahy Zsuzsa
					Peidl Máttyás Tibor	9.	
					Pósa Tamás	9.	
23-25.	zsokada	15	Lovassy László Gimnázium	Veszprém	Eszlári Dávid	10.	Pozsgainé Becze Boglárka
					Jáger Zsombor	10.	

## Dusza Árpád Országos Programozói Emlékverseny

					Bognár Katalin	10.	
23-25.	Gigabyte-keg- tők	15	Lovassy László Gimnázium	Veszprém	Lakatos András	9.	BáderAnikó Szabó László
					Hardi Roland	9.	
					Réti Péter	9.	
26.	ASD	9	Szolnoki Műszaki Szakképzési Cent- rum Pálffy-Vízügyi Szakgimnáziuma	Szolnok	Erdei Barnabás	10.	Englert Ervin
					Molnár Dániel	10.	
					Báthly Ádám	10.	
27.	xyz-123	8	Kőbányai Szent László Gimná- zium	Budapest	Berecz Dániel	9.	Nagy Tibor Imre
					Besenyi Dávid	9.	
					Nagy Bálint	9.	
28.	Xprogrammers	5	Városmajori Gimnázium és Kós Károly Áltá- lános Iskola	Budapest	Áprily Gergő	10.	Bárnkopf Bence
					Szerető Benedek	10.	
					Pongrácz Vince	10.	
29.	sajtok	4	Szolnoki Műszaki Szakképzési Cent- rum Pálffy-Vízügyi Szakgimnáziuma	Szolnok	Elekes Zsolt	10.	Englert Ervin
					Palya Patrik	10.	
					Jenei Gábor	10.	

Regionális forduló (hagyományos verseny)

11-13. évfolyam

	A csapat jellegje	A regionális fordulóban elért pontszám	Iskola	Az iskola székhelye	Csapattagok		Felkészítő tanárok
I.	Rendszerhiba újratöltve	114	Földes Ferenc Gimnázium	Miskolc	Robotka István Adrián	11.	Csató Endre
					Gáspár Attila	11.	
					Pál Diána	10.	
II.	Padászok	110	Budapesti Fazekas Mihály Általános Iskola és Gimnázium	Budapest	Alexy Marcell	11.	Pásztor Attila Weisz Ágoston Szoldatics József
					Szakály Marcell	11.	
					Németh Balázs	11.	
III-IV.	CoffeeConversion DethMachine	108	Kőbányai Szent László Gimnázium	Budapest	Csuhai Győző	11.	Nagy Tibor Imre
					Fekete Ádám Zsolt	11.	
					Szalai Patrik	11.	
III-IV.	for(int	108	Földes Ferenc Gimnázium	Miskolc	Sillinger Péter	12.	Csató Endre
					Bagány Viktor	12.	
					Lencsés Ádám	12.	
V.	Bagoly	104	Budapesti Fazekas Mihály Általános Iskola és Gimnázium	Budapest	Molnár-Sáska Zoltán Gábor	11.	Pásztor Attila Weisz Ágoston Nikházy László Dankovics Attila
					Gergely Patrik	11.	
					Janzer Orsolya Lili	11.	
VI.	köbgyök	102	Sárospataki Árpád Vezér Gimnázium és Kollégium	Sárospatak	Kucsinka Kristóf	12.	Szedes László
					Leitner Csaba	12.	
					Schmiedt Balázs András	12.	
7.	640kB	96	Szolnoki Műszaki Szakképzési Centrum Pálffy-Vízügyi Szakgimnáziuma	Szolnok	Gál József Patrik	11.	Englert Ervin Kovács Gergely Péter
					Kovács Luca Ágota	11.	
					Németh Ferenc	11.	
8.	Brassó2	95	Árpály Lajos Főgimnázium	Brassó	Longauer Zsolt	XI.	Bálint Ferenc, Vrencian Éva
					Nagy Edward Szilárd	XI.	
					Soos Márton	XI.	
9-11.	Halt and Catch Fire	92	Budapesti Műszaki Szakképzési Centrum Neumann János Számítástechnikai Szakgimnáziuma	Budapest	Princz Tamás	12.	Répásné Babucs Hajnalka
					Petróczi Zoltán	12.	
					Farkas Dániel	12.	
9-11.	vörösfenyő	92	Tóth Árpád Gimnázium	Debrecen	Vigh Franciska Fanni	12.	Kelemen Mihály
					Hermann Csaba	12.	
					Murvai András	12.	
9-11.	MÁJKELLSZAFT	92	Szolnoki Műszaki Szakképzési Centrum Pálffy-Vízügyi Szakgimnáziuma	Szolnok	Mészáros Dániel	12.	Englert Ervin Tóth Tivadar
					Juhász Kristóf	12.	
					Palik Bence	12.	
12-13.		90		Budapest	Győri Ferenc Norbert	12.	Nagy Tibor Imre

Döntőbe jutottak

	Második Rendszánsz		Kőbányai Szent László Gimnázium		Bakulár Márk Mihály	12.	
					Molnár Levente	12.	
12-13.	csk69	90	Kőbányai Szent László Gimnázium	Budapest	Kecskés Péter	12.	Nagy Tibor Imre
					Nagy Balázs	12.	
					Gidófalvi Márk	12.	
14.	Sóstalanság	85	Budapesti Műszaki Szakképzési Centrum Neumann János Számítástechnikai Szakgimnáziuma	Budapest	Dobrina Márton	12.	Répásné Babucs Hajnalka
					Mészáros Szabolcs	12.	
15.	Sok hűhó semmiért	82	Debreceni Szakképzési Centrum Mechwart András Gépipari és Informatikai Szakgimnáziuma	Debrecen	Bodó István	12.	Péter Miklós Kertész János Andrásné
					Bikfalvi Balázs	12.	
					Toma Armand	12.	
16-17.	Pepe The Frog	80	Lovassy László Gimnázium	Veszprém	Csarmasz Balázs	12.	Báder Anikó
					Koch Viktor	12.	
					Mayer László	12.	
16-17.	fsociety	80	Lovassy László Gimnázium	Veszprém	Balogh Ákos	12.	Báder Anikó
					Kazal Soma	12.	
					Torkos Csongor	12.	
18.	KEKSZ	76	Bartók Béla Elméleti Líceum	Temesvár	Orosz Kelemen	XI.	Mauzer Erika
					Szász Tamás	XI.	
					Bokor Zalán-Béla	XI.	
19.	TOB	75	Sárospataki Árpád Vezér Gimnázium és Kollégium	Sárospatak	Molnár András Imre	11.	Szeder László
					Leczó Gábor Bálint	11.	
					Szabó Martin	11.	
20.	Feláldozhatók	74	Városmajori Gimnázium és Kós Károly Általános Iskola	Budapest	Huszár Csaba	12.	Bárnkopf Bence
					Kerekes Áron	12.	
					Solymossy Ákos	12.	
21-22.	IPv5	65	Debreceni Szakképzési Centrum Mechwart András Gépipari és Informatikai Szakgimnáziuma	Debrecen	Agárdi Balázs	11.	Péter Miklós Rác Csaba
					Kóti Bence	11.	
					Nyika Benedek	11.	
21-22.	Agita Bevi	65	Debreceni Szakképzési Centrum Mechwart András Gépipari és Informatikai Szakgimnáziuma	Debrecen	Kerekes Ádám Koppány	11.	Rác Csaba
					Svélec Péter	11.	
					Vágner Dániel	11.	
23.	Another_bad_idea	63	Kecskeméti Szakképzési Centrum Kándó Kálmán Szakgimnáziuma és Szakközépiskolája	Kecskemét	Jaksa Márkó Dániel	11.	Klein Péter Róka Sándor
					Jakab Dávid	11.	
					Varga Bence	11.	
24-25.	Alma	60	Szent István Gimnázium	Budapest	Sillye Martin Zoltán	11.	Borbásné Penke Judit Szalayné Tahy Zsuzsa
					Török Gergely Balázs	11.	

## Dusza Árpád Országos Programozói Emlékverseny

					Báskay Dávid	11.	
24-25.	Program Ninják	60	Debreceni Szakképzési Centrum Mechwart András Gépipari és Informatikai Szakgimnáziuma	Budapest	Vida Imre	10.	Fekete Balázs Varga Sándor
					Nagy Dániel	14.	
					Nagy Erzsébet	14.	
26-28.	Deus Vut	55	Táncsics Mihály Gimnázium	Kaposvár	Fekete-Páris Ádám	11.	Kovács Ágnes Raskoványi Miklós
					Hajdu Marcell	11.	
					Rózsa Bálint	11.	
26-28.	refi.megoldas();	55	Kisvárdai Református Óvoda, Általános Iskola, Gimnázium és Kollégium	Kisvárdai	Borkó Károly	11.	Rolyák Géza
					Potyók Csaba	9.	
					Takács Bence	10.	
26-28.	KOLCSEY	55	Zalaegerszegi Kólcsey Ferenc Gimnázium	Zalaegerszeg	Buruczky Viktor József	11.	Henczi Béla
					Bödey Károly	12.	
					Takács Bendegúz	11.	
29.	Brassó1	52	Áprily Lajos Főgimnázium	Brassó	Hammas Attila	XII.	Bálint Ferenc, Vrencian Éva
					Molnár Szabolcs	XII.	
					Puskás-Bajkó Kamill	XII.	
30.	calcexe	45	Iryni János Református Szakgimnázium, Szakközépiskola és Diákotthon	Kazincbarcika	Balla Tamás	12.	Borsodi Csaba Kasza László Róbert
					Takács Dániel	11.	
					Vass Tamás	9.	
31.	Gőzmozdonyra	43	Árpád-házi Szent Erzsébet Középsiskola, Óvoda és Általános Iskola	Esztergom	Tókos Péter	12.	Hidvégi Anikó
					Wild János	12.	
					Pajor Gergő	12.	
32.	Not Python	40	Szolnoki Műszaki Szakképzési Centrum Pálffy-Vízügyi Szakgimnáziuma	Szolnok	Barát Zoltán	12.	Englert Ervin Tóth Tivadar
					Bathó Kristóf	12.	
					Munkácsi Zoltán	12.	
33-34.	Bencék	35	Kőbányai Szent László Gimnázium	Budapest	Kapoli Bence Kristóf	12.	Nagy Tibor Imre
					Baranyai Bence Dominik	12.	
33-34.	csapatnév	35	Kecskeméti Református Gimnázium	Kecskemét	Somogyi Soma	9.	Bodor Zoltán
					Tuska József	12.	
35.	Keri	20	V. István Katolikus Szakközépiskola és Gimnázium, Árpád-házi Szent Margit Általános Iskola	Sátoraljaújhegy	Vercs Botond	12.	György Éva
					Butkai Richárd	13.	
					Butella Bence	13.	

**Döntő (hagyományos verseny)**

**II. kategória (11-13. évfolyam)**

	A csapat jeligéje	Az elért pontszám	Iskola	Az iskola székhelye	Csapattagok		Felkészítő tanárok
I.	Rendszerhiba újratöltve	195	Földes Ferenc Gimnázium	Miskolc	Robotka István Adrián	11.	Csató Endre
					Gáspár Attila	11.	
					Pál Diána	10.	
II.	Padászok	181	Budapesti Fazekas Mihály Általános Iskola és Gimnázium	Budapest	Alexy Marcell	11.	Pásztor Attila Weisz Ágoston Szoldatics József
					Szakály Marcell	11.	
					Németh Balázs	11.	
III.	Bagoly	163	Budapesti Fazekas Mihály Általános Iskola és Gimnázium	Budapest	Molnár-Sáska Zoltán Gábor	11.	Pásztor Attila Weisz Ágoston Nikházy László Dankovics Attila
					Gergely Patrik	11.	
					Janzer Orsolya Lili	11.	
III.	kőbgyök	163	Sárospataki Árpád Vezér Gimnázium és Kollégium	Sárospatak	Kucsinka Kristóf	12.	Szeder László
					Leitner Csaba	12.	
					Schmiedt Balázs András	12.	
IV.	for(int	158	Földes Ferenc Gimnázium	Miskolc	Sillinger Péter	12.	Csató Endre
					Bagány Viktor	12.	
					Lencsés Ádám	12.	
IV.	CoffeeConversion DethMachine	158	Kőbányai Szent László Gimnázium	Budapest	Csuhai Győző	11.	Nagy Tibor Imre
					Fekete Ádám Zsolt	11.	
					Szalai Patrik	11.	

**I. kategória (8-10. évfolyam)**

	A csapat jeligéje	Az elért pontszám	Iskola	Az iskola székhelye	Csapattagok		Felkészítő tanárok
I.	Hókuszpokusz	158	Szent László Katolikus Gimnázium, Szakgimnázium, Általános Iskola, Kollégium és Óvoda	Kisvárd	Molnár Máté	8.	Dajka Miklós
					Jedla Martin	8.	
					Kovács Marcell	9.	
III.	nosleepnolife	132	Kőbányai Szent László Gimnázium	Budapest	Rádai Ronald	9.	Nagy Tibor Imre
					Szedlák Bence Patrik	9.	
					Vörös Asztrik	9.	

II.	Bitmozgatók	126	Esztergomi Dobó Katalin Gimnázium	Esztergom	Szén Máté Gábor	9.	Szurdi Gábor
					Zrupkó Zoltán	9.	
					Szerencsés Dániel	9.	
IV.	BACKSLASCH	124	Nyugat-magyarországi Egyetem Bolyai János Gyakorló Általános Iskola és Gimnázium	Szombathely	Schütz Brúnó	9.	Dobre Norbert
					Poór Máté	9.	
					Dobos Dominik	9.	

A verseny végeredménye a döntőben szerzett pontszám (max. 150 pont) és a regionális fordulóban elért pontszám felének összege alapján alakult ki.

Dusza Árpád családja 2012-ben különdíjat alapított, amelyet az idei tanévben a BACKSLASCH csapat kapott meg.

Az SAP a verseny támogatójaként az idei évben két csapat számára ajánlott fel különdíjat. tölthet el a SAP-nál

A Rendszerhiba újrátöltve és a Padászok csapata egy egész napot tölthet el a SAP-nál.

A legjobb csapatok tagjai, ha az ELTE-n szeretnének informatikát tanulni, az elért eredményük alapján pályázattal bekerülhetnek a **tehetségondozó képzésbe**.



Web-mobil verseny

		Az elért pontszám	Iskola	Az iskola székhelye	Csapattagok		Felkészítő tanár(ok)	
Web programozás	I.	244	Szegei Radnóti Miklós Kísérleti Gimnázium	Szeged	Fazekas Márk	11.	Gutai Árpád Tamás	
					Kaposi Benedek	11.		
					Horváth János	11.		
	II.	175	Zalaegerszegi Kölcsey Ferenc Gimnázium	Zalaegerszeg	Buruczky Viktor József	11.	Henczi Béla	
					Bódei Károly	12.		
					Lemle Béla	11.		
	III.	141	Budapesti Műszaki Szakképzési Centrum Neumann János Számítástechnikai Szakgimnáziuma	Budapest	Zászlós Márton György	11.	Répásné Babucs Hajnalka Drávucz Katalin	
					Dunai Tamás	11.		
					Miklósi Ákos	12.		
	IV.	127	Kecskeméti Szakképzési Centrum Kandó Kálmán Szakgimnáziuma és Szakközépiskolája	Kecskemét	Tóth István	10.	Trepák Ildikó	
Ádám Kristóf					11.			
Nagy Gergő					11.			
V.	100	Kaposvári Táncsics Mihály Gimnázium	Kaposvár	Pete Dávid	10.	Biczóné Lengyel Beata, Kovács Ágnes, Raskoványi Miklós		
VI. I. forduló- ból	50	Kőbányai Szent László Gimnázium	Budapest		Győri Ferenc Norbert		12.	Nagy Tibor Imre
					Bakulár Márk Mihály		12.	
				Molnár Levente	12.			
Mobil programozás	I.	295	Sárospataki Árpád Vezér Gimnázium és Kollégium	Sárospatak	Leitner Csaba	12.	Szeder László	
					Kucsinka Kristóf	12.		
					Schmiedt Balázs András	12.		
	II.	249	Földes Ferenc Gimnázium	Miskolc	Gyulai Márton	10.	Csató Endre	
					Bidlek Márton	11.		
					Robotka Adrián	11.		
	III.	182	ELTE Radnóti Miklós Gyakorló Általános Iskola és Gyakorló Gimnázium	Budapest	Fuchs Gábor	10.	Tasnádi Ildikó	
					Halmi Kristóf	10.		
					J. Tóth Tamás	12.		

INFORMATIKA -SZÁMÍTÁSTECHNIKA TANÁROK EGYESÜLETE

1133 Budapest, Pannónia u. 72-74.

- tel/fax: 1/462-0415 •
- e-mail: [isze@isze.hu](mailto:isze@isze.hu)
- web: [www.isze.hu](http://www.isze.hu)

Az egyesület alapítási éve: 1991.

Felelős kiadó: Dr. Bánhidi Sándorné

Szerkesztő: Szécsiné Festő-Hegedűs Margit