

Programozási feladatok futtatása és ellenőrzése Moodle alatt

C

Bizonyos elemek a Szeghalmy Sz., Fazekas A., Moodle használata a programozás oktatásban, (MoodleMoot 2019, Debreceni Egyetem, Informatikai Kar, június 24–25.)-ből származnak.

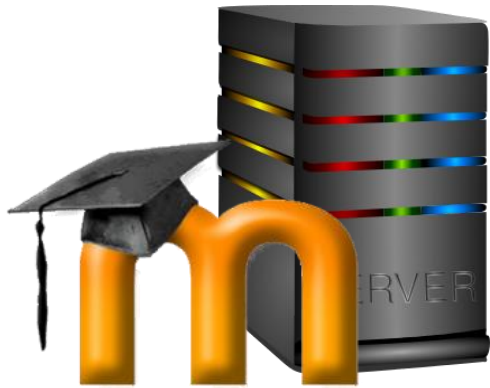
Tartalom

- ▶ A kérdéstípus elérése
- ▶ Egy programírási feladat
- ▶ Egy függvényírási feladat felületen át megadott tesztesetekkel
- ▶ Kiértékelő sablon módosítása: esetenkénti sablon
- ▶ Tesztesetek generálása és kiértékelés elve

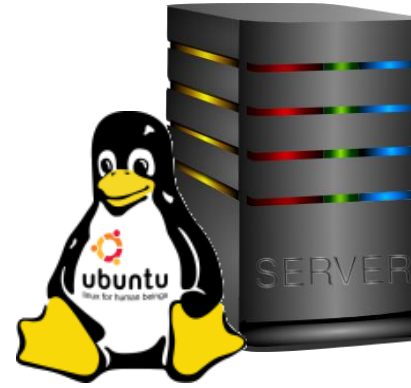
Megjegyzés

- ▶ A létrehozott feladatok ugyanúgy felhasználhatók tesztekben, mint pl. egy feleletválasztós kérdés.

A háttér



Moodle szerver
CodeRunner plug-in



A forráskódok futtatására
szolgáló szerver

Ubuntu 18.04.1 LTS

[job engine](#)

Jelenleg: gcc 7.5.0

Kérdés létrehozása

- ▶ Kérdéseket létrehozni egy adott kurzuson belül tud, válassza ki a megfelelő kurzust.
- ▶ A kurzusadminisztrációnál kattintson a kérdésbank gombra.

The screenshot shows the top navigation bar of an LMS. On the left, there are two icons: a gear (Settings) and a pencil (Edit). To the right of these icons is the text 'E-learning szolgáltatások' followed by a dropdown arrow, and 'Tájékoztató a távoktatási lehetőségekről'. Below the navigation bar, there are two main sections. The left section is titled 'Tevékenységek' (Activities) with a checkmark icon and the subtitle 'Kurzuselemek megtekintése' (View course elements). It contains two buttons: 'Fórumok' (Forums) and 'Tananyagok' (Materials). The right section is titled 'Kérdésbank' (Question Bank) with a question mark icon and the subtitle 'Tesztkérdések létrehozása és kezelése' (Creation and management of test questions). It contains two buttons: 'Kérdésbank' (Question Bank) and 'Kérdéskategória' (Question Category). A red arrow points from the gear icon in the navigation bar to the 'Kérdésbank' button in the right section.

Tipp:

A kérdések rendszerezése érdekében hozzon létre kategóriákat.

(Ez utólag is megtehető, a kérdések mozgathatók a kategóriák között is majd.)

Kérdés létrehozása

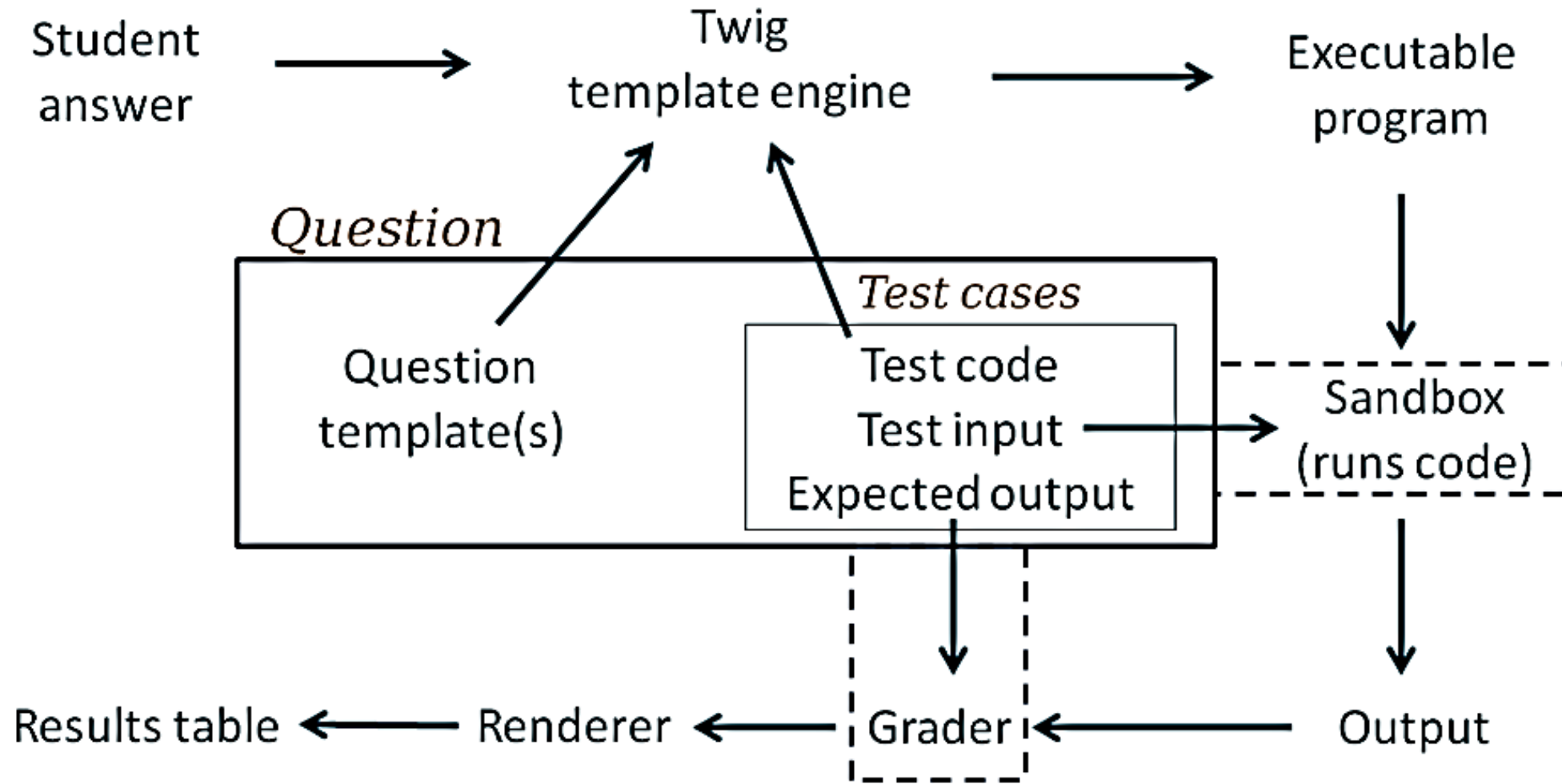
The screenshot shows the 'Kérdésbank' (Question Bank) interface. At the top, there are tabs for 'Kérdések', 'Kategoróriák', and 'Importálás'. Below the tabs, the title 'Kérdésbank' is displayed. A dropdown menu for 'Kategória kiválasztása:' is set to 'Bevezetés az infor...'. Below this, it says 'Nincsenek címkeszűrők'. There is a search filter dropdown 'Szűrés címkék szerint...'. Below the search filter, there are three checkboxes: 'A kérdés szövegének megjelenítése a ké...', 'Keresési feltételek', 'Alkategóriák kérdései is jelenjenek meg.', and 'A régi kérdések is jelenjenek meg'. At the bottom left, there is a button 'Új kérdés létrehozása...'. A red arrow points from this button to the 'CodeRunner' option in a dropdown menu on the right. The dropdown menu contains the following options: 'Párosító', 'Kiegészítendő kérdés', 'Számjegyes', 'Esszé', 'Beépített válaszos (kitöltő)', 'CodeRunner' (highlighted with a red 'Cr' icon), 'Egyszerű számításos', 'Elhúzás szövegbe', 'Elhúzható jelölők', and 'Képre húzás'.

- ▶ Kattintson az új kérdés létrehozására
- ▶ Válassza a CodeRunner kérdéstípust
- ▶ Adja hozzá a kérdést

A feladat alkotóelemei

- ▶ Beállítások
 - pl.: programnyelv, kiértékelés módja, kimenet megjelenítése
 - meghatározza a kérdéshez tartozó alapértelmezett sablont
- ▶ A kérdés
- ▶ A válasz
- ▶ Tesztesetek
 - Input adatok
 - Tesztelő kódrészek
 - Elvárt kimenet
- ▶ A sablon

Futtatás és kiértékelés



1. példa: C program

- ▶ Az input-output kezelése a feladat része
- ▶ Sablon: alapértelmezett
- ▶ Kiértékelés: Pontos egyezés szerint (alapértelmezés)
- ▶ Csak tökéletes megoldásra jár pont


Beállítások


CodeRunner question type
















Question type	<input type="text" value="c_program"/>	Az elvárt válasz egy teljes C program.
Customisation	<input type="checkbox"/> Customise <input type="checkbox"/> Template debugging	
Answer box	Rows <input type="text" value="18"/>	
Precheck	<input type="text" value="Disabled"/>	Ha nagyon sok a teszteset, érdemes átállítani.
Feedback	<input type="text" value="Force show"/>	
Marking	<input type="checkbox"/> All-or-nothing grading Penalty regime: <input type="text" value="0"/>	Csak hibátlanra kapjon pontot, vagy tesztesetenként jár részpont.

A kérdés szövege

- ▶ A kérdés nevének és szövegének megadása kötelező

A kérdés neve 

Kérdés szövege 

    **B** *I* U ~~S~~           

Készíts programot, mely beolvas a felhasználótól egy egész számot ($n > 0$) és kiszámolja az első n természetes szám összegét.

Kimenet formázása:

$S_n = \langle \text{eredmény_egesz_számként} \rangle$

Megj. A beolvasás során ne jelenítsd meg semmilyen üzenet.

Legyen előírása az input/output formázására.

A tesztesetek megadása

Test case 1	<input type="text"/>
Standard Input	<input type="text" value="5"/>
Expected output	<input type="text" value="Sn=15"/>

A feladat része a standard inputról történő beolvasás, ezért ki kell tölteni.

Javasolt az "Answer" mező kitöltése után generáltatni.

Test properties: Use as example Display Show Hide rest if fail Mark 1.000 Ordering 10

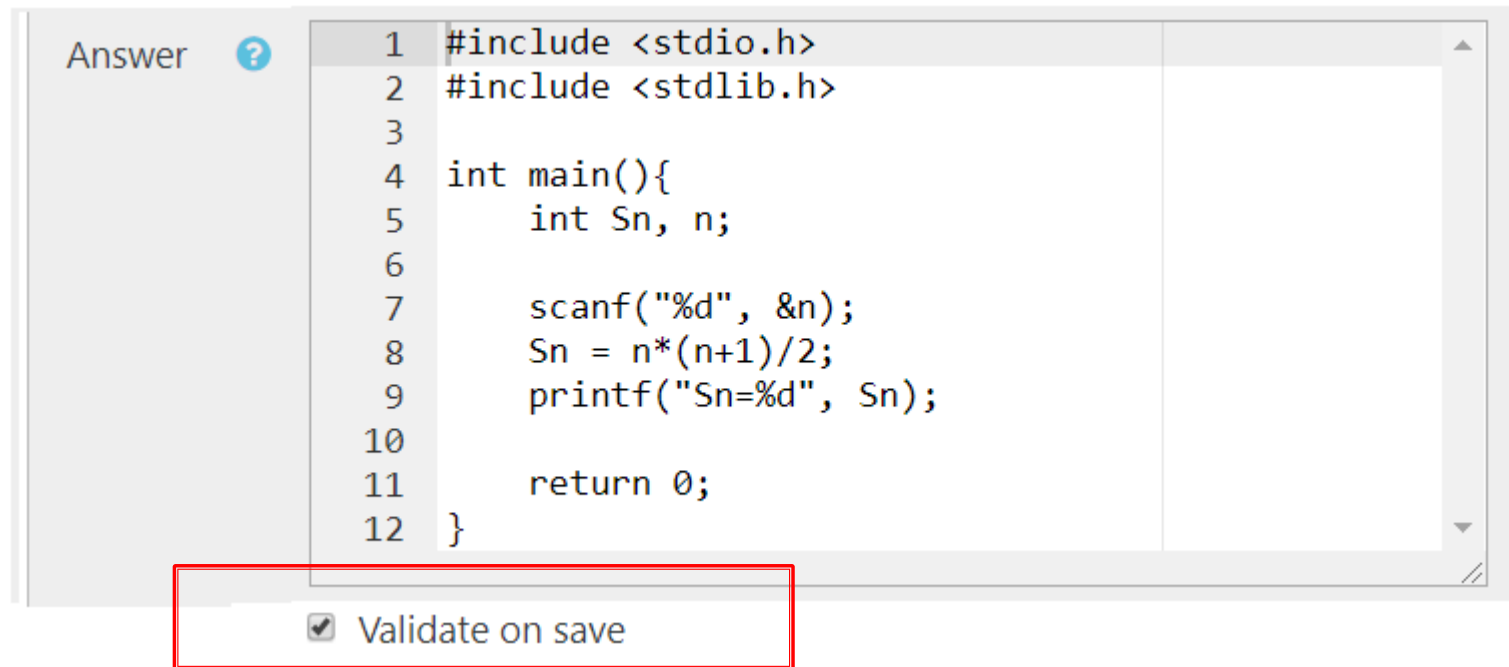
A kérdés szövege alatt példaként megjelenik a teszteset.

Elrejtjük teszteseteket (csalások megelőzése)

A helyes válasz megadása

▶ Ha megadjuk:

- automatikusan generálható lesz az elvárt válasz a tesztekhez
- a diákok a teszt beállításától függően láthatják ezt a megoldást



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int Sn, n;
6
7     scanf("%d", &n);
8     Sn = n*(n+1)/2;
9     printf("Sn=%d", Sn);
10
11     return 0;
12 }
```

Validate on save

Megj.: A feladatot érdemes készítés közben is menteni. A Validate on save-t csak akkor jelöljük be, amikor már kész a többi rész.

Az elvárt válaszok generálása

- ▶ Feladat mentése (a hiányzó válaszok miatt, sikertelen lesz)
- ▶ A felugró táblázatban kattintsunk a << jelekre
- ▶ Mentsük el. Kész a feladat.

Test	Expected	Got
Test case 1		Sn=15 <input type="button" value="<<"/>
Test case 2		Sn=3 <input type="button" value="<<"/>

Click on the << button to replace the expected output of this testcase with actual output.

For more detailed information, save the question with 'Validate on save' unchecked and test manually

A hallgató által látott feladatkiírás

Készíts programot, mely beolvassza a felhasználótól egy egész számot ($n > 0$) és kiszámolja az első n természetes szám összegét.

Kimenet formázása:

$S_n = \langle \text{eredmény_egesz_számként} \rangle$

Megj. A beolvasás során ne jelenítsd meg semmilyen üzenet.

For example:

Input	Result
5	$S_n = 15$
2	$S_n = 3$

A feladat
szövege

A példaként megjelölt
tesztesetekből
automatikusan generált
táblázat

Egy megoldási kísérlet

- ▶ A hallgató a válasz mezőbe beírhatja a megoldását
- ▶ Az Ellenőrzés gombra kattintva kérheti a kiértékelést
- ▶ A teszt beállításától függően azonnal vagy később láthatja az eredményét

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int n;
6     scanf("%d", &n);
7     printf("Sn=%d", n+1*n/2);
8     return 0;
9 }
```

Ellenőrzés

	Input	Expected	Got	
✘	5	Sn=15	Sn=7	✘
✔	2	Sn=3	Sn=3	✔
✘	17	Sn=153	Sn=25	✘
✘	53	Sn=1431	Sn=79	✘
✘	23	Sn=276	Sn=34	✘

Some hidden test cases failed, too.

Értesül arról, hogy legalább egy rejtett tesztesetre is hibás a válasza.

2. példa: C függvény

- ▶ Egy az előbbinél szigorúbb pontozási beállítást láthatunk
 - *All-or-nothing grading*: indokolt lehet gyakorlásnál és ZH-nál is
 - *Penalty regime*: csak az „álmából felébresztve is tudni kell” kérdésekhez

The screenshot shows a configuration panel for a question. The 'Question type' dropdown is set to 'c_function'. The 'Marking' section has 'All-or-nothing grading' checked and 'Penalty regime' set to '0,5,10,...'. Other settings include 'Customisation' (Customise and Template debugging unchecked), 'Answer box' (Rows: 14), 'Precheck' (Disabled), and 'Feedback' (Force show).

Callout 1: A hallgató csak a függvényt írja (általában annak is csak a törzsét)

Callout 2: Csak a tökéletes megoldásra kap pontot.

Callout 3: Hibás beküldések után járó levonás százalékban. A ... a két utolsó szám alapján egészíti ki a tételleket 100-ig.

A kérdés szövege

- ▶ Adjuk meg a feladat szövegét
- ▶ Töltsük ki a válasz mezőt (ha generáltatni akarjuk az elvárt válaszokat)
- ▶ Adjuk meg a függvény fejlécét az *Answer box preload* mezőben
A fejléc hiányában a feladat kiírásának minden ehhez szükséges információt tartalmaznia kell.

```
Answer box preload ?  
1 int feladat(int csokikSzama[], int n, int d){  
2  
3  
4  
5     return -1;  
6 }
```

Ebbe a dobozba egyéb segítő kódrészletek vagy kommentek is kerülhetnek.

Függvény: a tesztesetek megadása

Test case 1	<pre>int t[]={1, 2, 5}; int d = 10; printf("%d", feladat(t, sizeof(t)/sizeof(t[0]), d));</pre>	A t tömb és a d érték változik majd a teszteseteknél.
Standard Input		Ha a feladat nem vár el adatbekérést, akkor üresen marad.
Expected output	-1	A korábban látott módon generáltathatjuk, nem kötelező kézzel megadni.

- ▶ A tesztesetek megadása után mentjük az tesztet.
- ▶ Ha az Expected output még nem volt kitöltve/hibás volt, javítsuk és mentjük újra.

Mikor változtassunk a sablonon?

- ▶ Válasz mező tartalmának elemzésére
 - „Old meg while ciklussal...”,
 - „A megoldás során ne használd a math modult...”
- ▶ A kiértékelés megváltoztatására
 - részpontok osztása
 - **közelítő megoldások elfogadása** $\text{abs}(\text{exp} - \text{got}) < \text{eps}$
- ▶ Tesztetek generálására
 - véletlen értékekkel való tesztelés
 - ...

A legegyszerűbb sablonok (c function)

Tesztesetenkénti

```
#include ... // különböző include-ok  
#define SEPARATOR "#<ab@17943918#@>#"
```

```
{{ STUDENT_ANSWER }}
```

```
int main(){  
{{ TEST.testcode }}  
};
```

Egy konkrét
teszteset kerül be a
programba

A válasz mező tartalma
szövegszerűen
behelyettesítődik

Az összes teszteset
bekerül a
programba

Kombinált

```
#include ... // különböző include-ok  
#define SEPARATOR "#<ab@17943918#@>#"
```

```
{{ STUDENT_ANSWER }}
```

```
int main(){  
    {% for TEST in TESTCASES %}{  
        {  
            { TEST.testcode }};  
        }  
        {% if not loop.last %}  
            printf("%s\n", SEPARATOR);  
        {% endif %}  
    {% endfor %}  
    return 0;  
}
```

3. Példa: Közelítő egyezés figyelése

- ▶ Tesztesetenkénti sablont fogunk használni
 - IsCombinator: False
- ▶ Az általunk megadott sablon végzi majd a pontozást:
 - Template grader


Question type	<input type="text" value="c_function"/>
Customisation	<input checked="" type="checkbox"/> Customise <input type="checkbox"/> Template debugging
Answer box	Rows <input type="text" value="10"/>
Precheck	<input type="text" value="Disabled"/>
Feedback	<input type="text" value="Force show"/>
Marking	<input type="checkbox"/> All-or-nothing grading Penalty regime: <input type="text"/>

<input checked="" type="checkbox"/> Is combinator	<input type="checkbox"/> Allow multiple stdins	Test splitter (regex)	<input type="text" value=" #<ab@17943918#@>#\n ms"/>
<input type="text" value="Template grader"/>			
<input type="text" value='[["Input", "input"], ["Got", "got"], ["Expected", "expected"], ["Mark", "awarded"]]'/>			

Az eredmény tábla fejléce


Válasz mezők

- ▶ Adjuk meg a feladat szövegét és töltjük ki a válasz mezőket
E példában egy szám reciprokát kell a hallgatónak meghatározni.

Answer 

```
1 //nem kötelező megadnunk
2 double feladat(int a){
3     return 1.0/a;
4 };
```

Validate on save

Answer box preload 

```
1 double feladat(int a){
2
3 };
```

A tesztesetek megadása

- ▶ Adjuk meg a tesztesetet. A sablonban szereplő kódrészlettel összhangban kell lennie.
- ▶ Pl.:

Test case 1 ?

Sablon:

```
int a = {{ TEST.testcode }};
```

Test case 1 ?

Sablon:

```
{{ TEST.testcode }};
```

- ▶ Ha az elvárt választ is megadjuk, arra a `{{TEST.expected}}` néven hivatkozhatunk a sablonban.

A pontozást végző sablon

- ▶ Meghívja a hallgató által írt függvényt
- ▶ Összeveti az eredményt és az elvárt eredményt
- ▶ Kiírja a visszajelzést tartalmazó táblázat egy sorát a standard outputra.

- ▶ A táblázat szokásos oszlopai:
 - Iscorrect: helyes, helytelen, részben helyes (vizális visszajelzés)
 - Input: a tesztadat
 - Got: a diák megoldásának eredménye
 - Expected: az elvárt eredmény
 - Fraction: a tesztesre kapott pont

A sablon (esetenkénti)

```
//include állományok
```

```
{{ STUDENT_ANSWER }}
```

```
int main() {
```

```
    int a = {{ TEST.testcode }};
```

```
    double got = (double)feladat(a);
```

```
    double exp = 1.0/a;    //elvárt válasz számítása
```

```
    double res = fabs(got-exp)<EPS ? 1.0 : 0.0;
```

```
    printf("{\\"iscorrect\\":%0.1f, \\"input\\":%d, \\"got\\":%f, \\"expected\\":%f, \\"fraction\\":%f}",  
           res, a, got, exp, res);
```

```
    return 0;
```

```
}
```

Azért nem random, mert úgyis ki kell tölteni a teszteset részt a felületen. (a kísértés nagy...)

1: helyes
0: helytelen
0<iscorrect<1: részben helyes

A saját sablon a diák számára nehezen értelmezhető hibákhoz vezethet. (pl.: a diák függvénye nem ad vissza értéket, vagy rossz típusú értéket ad). Kifinomultabb megoldások erre is figyelnek.

Az aktuális tesztesetre adott pontszám

A visszajelzés

A táblázat oszlopai a sablonban szereplő printf utasítás és a felületen megadott fejlécek alapján állnak össze.

	Input	Got	Expected	Mark	
✓	10	0.1	0.1	1	✓
☑	5	0.1	0.2	0	☑
☑	6	0.1	0.166667	0	☑

„All-or-nothing grading” beállítással

	Input	Got	Expected	Mark	
✗	10	0.2	0.1	0	✗
✓	5	0.2	0.2	1	✓
✗	6	0.2	0.166667	0	✗

„All-or-nothing grading” nélkül

A kombináló sablon

- ▶ Létrehozza a visszajelzést tartalmazó táblázat fejlécét
- ▶ Generálja a tesztesetet
- ▶ Minden tesztesetre:
 - kiértékeli a hallgató megoldását
 - kiszámolja a helyes eredményt
 - összeveti az eredményt és az elvárt eredményt
 - hozzáfüzi az adott tesztesetre vonatkozó sort a táblázathoz
- ▶ Kiírja az **összeredményt** és a **visszajelző táblázatot** a standard outputra
- ▶ A legritkébb esetben készül C-ben (Python vagy C++ kérdéstípussal szokás megoldni)
- ▶ Egy fapados kód látható a feltöltött Moodle XML állományban. Importálás után szebb kivitelben.