

„A” melléklet Programozási környezet

A csapatok számára rendelkezésre álló munkaállomásokon Windows és Linux operációs rendszerek, valamint az alábbi szoftverek lesznek elérhetők:

Windows alatt:

1. 64 bites Windows 8.1 Enterprise operációs rendszer
2. Internet Explorer, Mozilla Firefox és Google Chrome böngészők
3. Far Manager 3.0 és Total Commander 8.52a fájlkezelők
4. Notepad++ 6.9.2 szövegszerkesztő
5. Java SE 8 Update 101
6. MinGW gcc és g++ 4.9.4
7. Netbeans 8.1 (Java és C/C++ támogatással, MinGW C/C++ fordítóval)
8. Eclipse 4.6.0 Neon (Java és C/C++ támogatással)
9. Visual Studio Community 2013 (Java, C/C++ és C# támogatással, .NET 4.5 keretrendszerrel)
10. Code::Blocks 16.01 (C/C++ támogatással)
11. Free Pascal 3.0.0
12. Python 3.5.2

Linux alatt:

1. 64 bites Ubuntu 14.04 LTS Linux operációs rendszer
2. Gnome grafikus felhasználói felület
3. Mozilla Firefox böngésző
4. Midnight Commander fájlkezelő
5. mcedit, joe, kate, xemacs szövegszerkesztők
6. Java SE 8 Update 25
7. xwpe IDE
8. gdb nyomkövető ddd grafikus felülettel
9. gcc és g++ 4.8.4
10. Netbeans 8.1 (Java és C/C++ támogatással)
11. Eclipse 4.6.0 Neon (Java és C/C++ támogatással)
12. Code::Blocks 16.01 (C/C++ támogatással)
13. Free Pascal 3.0.0
14. Python 3.4.3

Az egyes nyelvek dokumentációi (C standard könyvtár, C++ STL, Java SE API, Free Pascal, Python 3) a verseny közben elérhetők lesznek. Felhívjuk a figyelmet arra, hogy a C# nyelvhez jelenleg nem tudunk dokumentációt biztosítani.

Kiemelt támogatóink:



Morgan Stanley

További támogatók:



„B” melléklet

A versenyen használható programozási nyelvekről, fordítóprogramokról és a forráskódok lefordításához használt fordítási opciókról, valamint technikai specifikációkról

1. A verseny programozási nyelvei: Java 8, C, C++11, C#, Pascal és Python 3.
2. Egy feladat megoldását egyetlen, a forrásnyelven megírt kódot tartalmazó állományban (továbbiakban forráskód) kell beküldeni. A beküldött állomány nevének az alábbi kiterjesztések egyikével kell rendelkeznie:
 - Java nyelvű program esetén: `.java`
 - C nyelvű program esetén: `.c`
 - C++ nyelvű program esetén: `.cpp`
 - C# nyelvű program esetén: `.cs`
 - Pascal nyelvű program esetén: `.pas`
 - Python nyelvű program esetén: `.py`
3. A versenyzők által beküldött forráskódok az alábbi parancsokkal lesznek lefordítva:
 - Java (Java 8 Update 25):
`javac -encoding UTF-8 <forrás.java>`
 - C (gcc 4.8.1):
`gcc -std=c99 -Wall -lm -DONLINE_JUDGE -O3 <forrás.c>`
 - C++ (g++ 4.8.1):
`g++ -std=c++11 -Wall -lm -DONLINE_JUDGE -O3 -msse2
-static-libstdc++ -s <forrás.cpp>`
 - C# (csc 12.0):
`csc <forrás.cs> /define:ONLINE_JUDGE /optimize`
 - Pascal (Free Pascal 3.0.0):
`fpc -DONLINE_JUDGE -XS -O2 <forrás.pas>`
 - Python (3.5.2):
`python -m py_compile <forrás.py>`
4. A beküldött forráskód mérete nem haladhatja meg a 40 KB-ot.
5. A beküldés forráskódjából lefordított futtatható állomány mérete nem haladhatja meg az 5 MB-ot.
6. Minden programnak 512 MB memória áll rendelkezésére a futása során.
7. A feladat megoldását végző programnak a befejeződésekor zéró (0) visszatérési értéket kell adnia az operációs rendszer felé (lásd a „C” mellékletet), ellenkező esetben „Futási hiba” értékelést kap a beküldés.
8. A beküldött forráskód nem tartalmazhat olyan kódrészleteket, amelyek
 - a standard input és standard output csatornákon kívül más állományokat használnak,
 - egynél több szálat vagy folyamatot kezelnek,
 - hálózati szolgáltatásokat vesznek igénybe.

Kiemelt támogatóink:



Morgan Stanley

További támogatók:



„C” melléklet

Programozási javaslatok a fordítási és futási hibák elkerülésére, illetve a programok futásának optimalizálására

A programkódokat beküldés előtt mindenképpen érdemes a „B” mellékletben megadott fordítási opciókkal lefordítani, elkerülendő a versenybizottság „Fordítási hiba” üzenetét.

A versenybizottság kiértékelő alkalmazásai figyelik a tesztelt beküldés által szolgáltatott visszatérési értéket. Ha ez az érték 0-tól különböző, a beküldés értékelése „Futási hiba” lesz. Egy program az operációs rendszer felé a következő módon tud 0 visszatérési értéket szolgáltatni:

- Java nyelven: **System.exit(0)**; metódushívással a program tetszőleges pontján, ahol végrehajtható utasítás állhat.
- C, C++ nyelven: **return 0**; utasítással a main() függvényben, illetve **exit(0)**; függvényhívással a program tetszőleges pontján, ahol végrehajtható utasítás állhat.
- C# nyelven: **Environment.Exit(0)**; metódushívással a program tetszőleges pontján, ahol végrehajtható utasítás állhat.
- Pascal nyelven: **Halt(0)**; eljáráshívással a program tetszőleges pontján, ahol végrehajtható utasítás állhat.
- Python nyelven: **sys.exit(0)**; függvényhívással a program tetszőleges pontján, ahol végrehajtható utasítás állhat.

A Java, C# és Pascal programokat nem kötelező a megadott utasításokkal befejezni, mivel azok – normál befejeződés esetén – amúgy is 0-val térnek vissza.

A C++ nyelven programozóknak javasoljuk, hogy a standard bemenetről történő adatbeolvasásra és a standard kimenetre történő adatkirratásra a program futásának optimalizálása érdekében a C++ nyelvben megszokott **cin** és **cout** streamek helyett használják a C nyelv standard könyvtári függvényeit, a **scanf()** és **printf()** függvényeket! A **cin** és **cout** streameket használó programok kellően nagyméretű adathalmazok esetén mérhetően lassabban futnak!

A versenybizottságnak a beküldések teszteléséhez használt adatai a feladatokban leírt input (bemeneti) specifikációknak megfelelőek lesznek. A beküldéseknek csak a specifikációban megadott formátumú adatokat kell feldolgozniuk. Nem szükséges a beküldött programban ezek helyességét ellenőrizni, ez ugyanis lassíthatja a program futását.

A program által a standard kimenetre küldött adatok formátumának meg kell felelnie a probléma output (kimeneti) specifikációjában leírtaknak. A versenybizottság kiértékelő alkalmazásai nem fogadják el helyesnek az olyan kimeneti eredményeket, amelyeknek a végén tetszőleges számú (üres) sorral több szerepel, mint az elvárt helyes kimenet, vagy éppen az eredmény valamely sora a kívánt helyes karaktersorozat után további szóköz karaktereket tartalmaz.

Különösen ellenjavallt a beküldött programban olyan, a felhasználónak szóló, ám a feladat kimeneti specifikációjában nem szereplő üzeneteknek a standard kimenetre történő kiírása, mint például a „Kérem a következő egész számot:” vagy az „Adja meg n és m értékét!” üzenetek. Ezek ugyanis a begépelte formában jelennek meg a standard kimeneten, és emiatt nagy valószínűséggel nem fognak megfelelni a feladat kimeneti specifikációjában leírtaknak.

Kiemelt támogatóink:



Morgan Stanley

További támogatók:





Regionális Programozó Csapatverseny Debrecen, 2016. december 4.



A programnak csak addig kell tárolnia a beolvasott bemeneti adatokat, ameddig azok feldolgozásra nem kerülnek; legtöbbször felesleges előre beolvasni az összes bemeneti adatot, és csak utána feldolgozni azokat. Legtöbbször ugyancsak felesleges a már kiszámított részeredményeket a memóriában tárolni, ha a további számításokhoz már nincsen szükség rájuk.

Ha a feladatnak a standard inputról kell adatokat beolvasnia, akkor a megoldást célszerű úgy tesztelni, hogy a tesztelésre szánt bemeneti adatokat előzetesen elmentjük egy szöveges állományba (`teszt.in`), majd egy parancsablakban kiadjuk a

```
program < teszt.in > teszt.out
```

parancsot. A `teszt.in` állomány létrehozásával a tesztelendő adatok ismételt begépelésétől kímélhetjük meg magunkat abban azt esetben, ha a program nem az elvárt eredményeket szolgáltatja valamely tesztesetre. A program lefutása után a kimeneti adatokat a `teszt.out` állományban találjuk.

Kiemelt támogatóink:



Morgan Stanley

További támogatók:

