

# 1. Whitespace

*Beküldendő: forráskód és rövid fejlesztői dokumentáció*

A feladatban a *Whitespace* programozási nyelv egy leegyszerűsített változatához kell értelmezőt írni. Míg a legtöbb nyelvben az úgynevezett whitespace-karaktereknek (a szóköznek, a tabulátornak és a sortörésnek) nincs hatása a program jelentésére, ebben a nyelvben éppen ellenkezőleg: az értelmező csak ezeket a karaktereket veszi figyelembe, minden más szimbólumot (betűket, számokat stb.) figyelmen kívül hagy. A nyelvet áprilisi tréfának szánták, de eredeti változatáról bizonyítható, hogy minden elvégezhető benne, ami C-ben, Pythonban vagy Javában is.

**Fontos:** az alábbiakban ismertetett nyelv hasonlít az „igazi” *Whitespace*-re, de annak erősen leegyszerűsített változata és sok mindenben eltér tőle; ezért az interneten elérhető dokumentáció félrevezető lehet! A feladat elvárásainak mindenképpen felelj meg, de ha van kedved, hozzáadhatsz egyéb funkciókat is.

## 1.1. A nyelv

Nyelvünkben tehát kizárólag három karakternek van jelentése: a tabulátornak, a szóköznek és a sortörésnek. Ezek kombinációjából állnak össze az utasítások, illetve esetenként az ezek paraméteréül adott számok. A további táblázatokban a szóközt [Space]-szel, a tabulátort [Tab]-bal, a sortörést [LF]-fel jelöljük.

Egy *Whitespace*-program futása során folyamatosan jelen van egy ún. *verem*. Ezt úgy kell elképzelni, mint „egymásra rakott” számokat. Az utasítások alapvetően a verem tetejét manipulálják (pl. a verem tetejére tesznek be egy számot, a verem tetején lévő számot távolítják el, a verem tetején levő két számot kicserélik az összegükre).

A feladatban a következő parancsokat kell megvalósítanod:

Parancs	Jelentés	Van paramétere?
[Space][Space]	A paraméterül kapott számot a verem tetejére rakja.	van
[Space][LF][Space]	A verem tetején levő számot még egyszer a verem tetejére rakja.	nincs
[Space][LF][LF]	A verem tetején levő számot eltávolítja.	nincs
[Tab][Space][Space][Space]	A verem tetején levő két számot kicseréli az összegükre.	nincs
[Tab][Space][Space][Tab]	Ugyanez kivonással. <i>Fontos:</i> a veremhez korábban hozzáadott szám lesz az első operandus, a kisebbítendő.	nincs
[Tab][Space][Space][LF]	Ugyanez szorzással.	nincs
[Tab][LF][Tab][Space]	Beolvas egy karaktert, és annak az ASCII-kódját a verem tetejére írja.	nincs
[Tab][LF][Tab][Tab]	Beolvas egy számot, és azt a verem tetejére írja.	nincs
[Tab][LF][Space][Space]	Kiírja azt a karaktert, aminek az ASCII-kódja a verem tetején levő szám. <i>Ez nem távolítja el a számot a verem tetejéről.</i>	nincs
[Tab][LF][Space][Tab]	Kiírja a verem tetején levő számot. <i>Ez sem távolítja el azt a verem tetejéről.</i>	nincs

Amelyik parancsnak paramétere van, az után egy számot is meg kell adni [Space]-ek és [Tab]-ok sorozataként, a következőképpen: az első karakter [Space], ha a szám nemnegatív és [Tab], ha negatív; utána pedig egy kettes számrendszerbeli felírás következik, ahol a [Space] a 0 és a [Tab] az 1. A leírást egy [LF]-nek kell zárnia. Például a [Space][Tab][Space][Space][Tab][LF] jelöli a +1001-et, vagyis a 9-et.

## 1.2. Példa

Az alábbi kód kiírja a „Hello world!” szöveget és egy sortörést a standard outputra.

```
S S S T S S T S S S L:+1001000=72-t,_a_’H’_karakter_ASCII-kodjat_a_verem_tetejere
T L
S S :kiirja_a_’H’-t;_S S S T T S S T S T L:+1100101=101=’e’-t_a_verem_tetejere
T L
S S :kiirja_az_’e’-t;_S S S T T S T T S S L:+1101100=108=’l’
T L
S S S S S T T S T T S S L:+1101100=108=’l’
```

```

T L
S S S S S T T S T T T L:+1101111=111='o'
T L
S S S S S T S T T S S L:+101100=44=', '
T L
S S S S S T S S S S S L:+100000=32=Space
T L
S S S S S T T T S T T T L:+1110111=119='w'
T L
S S S S S T T S T T T T L:+1101111=111='o'
T L
S S S S S T T T S S T S L:+1110010=114='r'
T L
S S S S S T T S T T S S L:+1101100=108='l'
T L
S S S S S T T S S T S S L:+1100100=100='d'
T L
S S S S S T S S S S T L:+100001=33='!'
T L
S S :kiirja_a_!'-t;_S S S T S T S L:+1010=10='\n'_(sortores)
T L
S S :kiirja_a_sortorest

```

Egy másik példakód beolvas két számot, majd az összegüket és egy sortörést kiír a standard outputra. *T*-t írtunk a tabulátor, *S*-t a szóköz és *L*-et a sortörés elé.

```

T L
T T :beolvas_egy_szamot|T L
T T :beolvas_meg_egy_szamot|T S S S :kicsereli_oket_az_osszegukre|T L
S T :kiirja_az_osszeguket|S S S T S T S L:+1010=10,_a_sortores_ASCII-kodja
|T L
S S :kiirja_a_sortorest

```

Utóbbinál a verem változása a futás során, ha pl. 4-et és 5-öt ad meg a felhasználó:

4				az első beolvasás
4	5			a második beolvasás
9				az összeadás
9				a kiírás (ez nem változtatja a vermet)
9	10			a 10 beszúrása a verem tetejére
9	10			a sortörés kiírása (ez sem változtatja a vermet)

### 1.3. A feladat

Egy programot kell írnod egy általad választott programozási nyelvben, amely a paraméterként kapott (jellemzően '.ws' kiterjesztésű) Whitespace-forrásfájlt értelmezi (vagyis sorban végrehajtja a megadott utasításokat). Emellett egy rövid (1-2 oldalas) dokumentációban foglald össze, hogyan működik a programod (milyen ötleteket alkalmaztál, hogyan strukturáltad a kódot stb.).

Hibakezeléssel nem kell foglalkoznod; feltételezheted, hogy a kapott kód helyes formátumú, és jelentésében is értelmes (azaz pl. nem próbál egy üres verem tetejéről olvasni). Előny, ha a hatékonyságra is figyelsz (minél gyorsabban, ügyesebben keresel az utasítások között), de ez sem kötelező.

## 1.4. Tippek

- Olvasd a karaktereket egymás után (közben a nem-whitespace szimbólumokat dobd el). Amint az eddigi karakterek egy értelmes utasítást adnak, hajtsd végre az utasítást (esetleg olvasd be a paraméterként kapott számot, ha van), majd kezd elölről.
- A legtöbb programozási nyelvben a tabulátort `'\t'`, a sortörést `'\n'` jelöli.
- Teszteld a megoldásod az általunk adott példakódokkal, illetve saját programokkal. Az interneten lévők nem feltétlen működnek, mert sok mindenben egyszerűsítettünk a nyelven.
- *Haladóknak:* ha hatékony kódot szeretnél írni, próbáld meg egy fastruktúrában ábrázolni a parancsokat, amelyek levelében a parancsoknak megfelelő függvényekre mutató pointerok vannak.
- Mikor hibát keresel, érdemes lehet a beolvasott karakterek magyarázatait, illetve az általuk megadott parancsokat kiírni. *(A végleges programban viszont már ne legyen fölösleges kiírás.)*
- Ha valami nem működik, nézd meg, hogy a Whitespace-kód nincs-e elírva; elég könnyű elírni...