

SZÖVEGKEZELÉS

GYAKORLÓ FELADATOK C#-BAN

SZTRING MŰVELETEK

A karakterlánc (string) típusú változókba szöveget olvashatunk be. Az ilyen változó maximális hossza 255 karakter.

STRING ÖSSZEFÜZÉSE

A sztring típuson értelmezett összefűzés operátor jele a '+'. A '+' operátor egy **kétooperandusú (bináris) operátor**, ez azt jelenti, hogy két operandusra van szüksége a művelet elvégzéséhez. A '+' operátor **infix operátor**, ami azt jelenti, hogy magát az operátort ('+' jel) a két operandus közé kell írni.

Ennek következtében helyes használata pl. "alma" + "fa", melynek eredménye egy új sztring, tartalmát tekintve a két sztring karaktereinek egyesített sorozata, jelen példa esetén "almafa".

STRING RÉSZÉNEK KIVÁGÁSA

A sztring egy részének kivágására a **Substring()** metódust használhatjuk. Pl. string s="almafa"; esetén string m=s.Substring(2,3); jelentése: vedd az s változóban lévő sztring második karakteréről kezdődő 3 karakter hosszú szakaszt.

Ehhez tudnunk kell, hogy a **karakterek sorszámozása** a sztring belsejében **0-val kezdődik**, vagyis a=0, l=1, m=2, a=3, f=4, a=5 sorszámmal szerepel. Így a szóban forgó szövegrész az "maf" karaktersorozat, ez kerül az m változóba.

Vigyázat! Amennyiben a SubString()-el olyan szövegszakaszra hivatkozunk, amelyik nem létezik, úgy futás közbeni hibát kapunk! Pl. s.Substring(4,6) esetén a 4-es karaktertől ('f') egy hat hosszú szakasz kellene, de nincs már annyi karakter hátra a sztringből. Ha csak egyetlen paramétert adunk meg a SubString() esetén, akkor azt jelenti, hogy a megadott pozíciótól minden további hátralévő karaktert ki kell vágni. Tehát az m=s.Substring(3); esetén az m értéke "afa" lesz.

SZTRING NAGYBETŰSÉ ALAKÍTÁSA

A sztring nagybetűsre alakítható a **ToUpper()** metódus segítségével. Pl. string s="almafára"; esetén string m=s.ToUpper(); jelentése: vedd az s változóban lévő sztringet, és alakítsd át minden karakterét nagybetűre. Így az m változó értéke "ALMAFÁRA" lesz.

Az átalakítás során a Windows nemzetközi beállításai kerülnek figyelembe véve, vagyis a fenti példa csak magyar nyelvű Windows-on fogja a fenti eredményt adni. Más nyelvre állított Windows esetén a magyar 'á' betű nem feltétlenül fog 'Á' alakra kerülni. **A nem ABC karakterek a nagybetűre alakítás során nem változnak meg.** Az m="3 medvebocs, 2 nyúl.".ToUpper(); hatására az m értéke "3 MEDVEBOCS, 2 NYÚL." lesz.

SZTRING KISBETŰSÉ ALAKÍTÁSA

A sztring kisbetűsre alakítható a **ToLower()** metódus segítségével. Pl. string s="ÁTMÁSZOTT az Árkádon!"; esetén string m=s.ToLower(); jelentése: vedd az s változóban lévő sztringet, és alakítsd át minden karakterét kisbetűre. Így az m változó értéke "átmászott az árkádokon" lesz.

Az átalakításra ugyanaz vonatkozik, mint ami a nagybetűs résznél olvasható.

SZTRING HOSSZÁNAK LEKÉRDEZÉSE

A sztring karaktereinek számát a **Length** tulajdonsággal kérdezhetjük le. Pl. `string s="almafa";` esetén `int l= s.Length;` jelentése: vedd az s változóban lévő sztringet, számold meg hány karakterből áll, és a számot helyezd el az l változóba. Így az l változó értéke 6 lesz.

A karakterszámba beleszámítódnak a szóköz jelek, valamint minden nem ABC-beli betű is. Az `s="Alma \n fa"` esetén az `s.Length` értéke 9 lesz (6 db betű, 2 db szóköz, 1 db speciális karakter, melynek a jele ugyan 2 db karakter ('\\n'), de valójában a 10-es kódú karaktert jelöli, így 1 karakternek számít). A sztring hosszába nem számít bele a kötelező, sztring végét jelző speciális lezáró karakter.

SZTRINGEK KARAKTERENKÉNTI OLVASÁSA

A sztring egy karaktersorozat, melynek elemei (karakterei) be vannak sorszámozva, 0-val kezdődően. A sztring egy elemét el tudjuk érni az indexével (sorszámaival). `string s = "almafa";` esetén a `char c = s[3];` értékadás során a sztring 3-as sorszámu karaktere kerül be a c változóba. Ez pedig az 'l' betű.

Vigyázat! Amennyiben a sztring olyan karakterére hivatkoznánk, amelyik nincs, úgy futás közbeni hibát fogunk okozni!

SZTRINGEK KEZDETÉNEK VIZSGÁLATA

A sztring elejét meg tudjuk vizsgálni az **StartsWith()** metódussal. Használata: `string s="almafa";` esetén például `if (s.StartsWith("alma")) ...`

A **StartsWith()** **logikai igaz vagy hamis értéket ad vissza** attól függően, hogy a sztring tényleg a megadott karaktersorozattal kezdődik-e. Ezen vizsgálat **kisbetű-nagybetű érzékeny**. Az `if (s.StartsWith("Almafa"))` például hamis, mivel nem nagy 'A'-val, hanem kis 'a'-val kezdődik.

SZTRINGEK VÉGÉNEK VIZSGÁLATA

A sztring végét meg tudjuk vizsgálni az **EndsWith()** metódussal. Használata: `string s="almafa";` esetén például `if (s.EndsWith("fa")) ...`

Az **EndsWith()** **logikai igaz vagy hamis értéket ad vissza** attól függően, hogy a sztring tényleg a megadott karaktersorozattal végződik-e. Ezen vizsgálat **kisbetű-nagybetű érzékeny**. Az `if (s.EndsWith("Fa"))` például hamis, mivel nem nagy 'F'-fel, hanem kis 'f'-fel szerepel benne a végén a "fa".

SZTRINGEK ÖSSZEHASONLÍTÁSA

Két sztring egyenlő akkor és csak akkor, **ha hosszuk egyenlő, és minden karaktere rendre egyenlő**. Két karakter egyenlő, ha mindkét karakternek egyforma a kódtábla szerinti kódja. Ennek megfelelően az "almafa" és az "alma" sztringek nem egyenlők (nem egyforma hosszúak). Az "almafa" és "Almafa" sem egyenlők, mivel az egyikben kis 'a', a másikban pedig nagy 'A' szerepel a nullás pozíción.

FELADATOK

1. KARAKTERSZÁM

Készítsünk programot, amely beolvas egy mondatot, majd kiírja, hogy hány karakterből áll a mondat.

2. RITKÍTOTT

Készítsünk programot, amely bekér egy mondatot, majd kiírja ugyanezt a mondatot úgy, hogy mindegyik betű (karakter) után kirak egy szóközt. Egy string típusú, **mondat** nevű változó i. karakterét így érhetjük el: `mondat[i]`.

3. NAGYBETŰS

Készítsünk programot, amely beolvas egy mondatot, majd kiírja a mondatot nagy betűkkel. A **mondat** nevű változóban lévő szöveg nagybetűssé alakítása: `mondat.ToUpper()`.

4. RITKÍTOTT NAGYBETŰS

Készítsünk programot, amely beolvas egy mondatot, majd kiírja a mondatot nagy betűkkel úgy, hogy mindegyik karakter után kirak egy szóközt. A **mondat** nevű változóban lévő szöveg i. karakterének nagybetűssé alakítása: `mondat[i].ToString().ToUpper()`.

5. ASCII KÓD → KARAKTER

Készítsünk programot, amely sorban kiírja az ASCII kódokhoz tartozó karaktereket a 0-ás kódtól a 255-ös kódig.

A program kimenete:

```
ASCII kód --> karakter program
Az ASCII karakterek:
0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~ ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ à á â ã ä å ç c é e ë ì í î ï ñ ò ó ô õ ö ÷ ù ú û ü ý ? y
```

6. KARAKTER → ASCII KÓD

Írassuk ki az „A” betűhöz tartozó ASCII kódot, majd mondjuk meg egy billentyűzetről tetszőlegesen beírt karakter ASCII kódját.

A program lehetséges kimenete:

```
karakter --> ASCII kód program
Az A karakter ASCII kódja: 65
Kérek egy karaktert: C
A(z) C karakter kódja: 67
```

7. FÜGGŐLEGES

Kérjünk be egy keresztnévet, majd írassuk ki betűnként függőlegesen lefelé a képernyőre.

A program egy lehetséges kimenete:

```
Függőleges program
Írj be egy keresztnévet: Pali

P
a
l
i
```

8. VISSZAFELÉ

Készítsünk programot, amely bekér egy mondatot, majd kiírja ugyanezt a mondatot visszafelé.

A program egy lehetséges kimenete:

```
Visszafelé program
Írj be egy mondatot: indul a görög aludni

A beírt mondat visszafelé:
indula görög a ludni
```

9. KARAKTERNÉGYZET

Olvassunk be egy **A** természetes számot és egy **CH** karaktert (char típusú). Rajzoljunk ki a beolvasott karakterből egy **A** oldalú négyzetet a képernyőre (minden sorban **A** db **karakter** legyen és összesen **A** db **sorunk** legyen) egymásba ágyazott ciklusok segítségével. A program A=4 és ch=a-ra az alábbi kimenetet adja:

```
Karakternégyzet program

Mennyi karakterből álljon a négyzet? 4
Milyen karakterekből álljon a négyzet? a

aaaa
aaaa
aaaa
aaaa
```

10. VOLT-E NAGYBETŰ

Készítsünk programot, amely bekér egy mondatot, majd kiírja, hogy volt-e benne nagybetű vagy sem.

Egy karakterről úgy tudjuk eldönteni, hogy nagybetű-e, hogy megnézzük nagyobb vagy egyenlő-e mint 'A' és kisebb vagy egyenlő-e mint 'Z'. Ha ez igaz, akkor a karakter A-tól Z-ig „van”, tehát egy nagybetű.

Fontos! A fenti módszer csak az angol ABC-ben szereplő karakterek esetén működik. A magyar ékezetes betűk a kódtáblában ettől hátrébb helyezkednek el, és nem közvetlenül egymás mellett vannak.

A program egy kimenete lehet az alábbi:

```
Volt-e nagybetű program

Írj be egy mondatot: Ez egy mondat.

A fenti mondatban volt nagybetű.
```

11. NAGYBETŰK SZÁMA

Egészítsük ki a fenti feladatot azzal, hogy ha van a mondatban nagybetű, akkor írja ki azt is, hogy hány nagybetű van benne.

A program egy kimenete lehet az alábbi:

```
Nagybetűk száma program
Írj be egy mondatot: Ez itt egy Nagybetűs mondat.
A fenti mondatban a nagybetűk száma: 2
```

12. NAGYBETŰK FELSOROLÁSA

Egészítsük ki a fenti feladatot azzal, hogy ha van a mondatban nagybetű, akkor ne csak azok számát írja ki, hanem sorolja fel azt is, hogy melyek voltak a nagybetűk.

A program egy kimenete lehet:

```
Nagybetűk felsorolása program
Írj be egy mondatot: Ebben a mondatban Van Több nagybetű Is.
A fenti mondatban ezek voltak a nagybetűk: E V T I
```

13. SZÓKÖZÖK SZÁMA

Kérjünk be egy mondatot. Számoljuk meg, majd írassuk ki, hogy hány szóköz van benne. A karakterek összehasonlítása azok ASCII kódja alapján történjen!

A program egy lehetséges kimenete:

```
Szóközők száma program
A mondat = Ez egy új mondat több szóközzel.
A fenti mondatban a szóközők száma: 5 db.
```

14. SZÓKÖZ NÉLKÜL

Kérjünk be egy mondatot, majd írassuk ki ugyanezt a mondatot szóközők nélkül. (Segítség: a szóköz karaktert az ASCII kódja segítségével találhatjuk meg a sztringben.)

A program egy lehetséges kimenete:

```
Szóköz nélkül program
A mondat = Több szóközből álló mondat.
Többszóközből álló mondat.
```

15. NÉV-KOMBINÁCIÓK

Kérjünk be két vezetéket- és két keresztnévet, majd írassuk ki az ezekből képezhető neveket!

A program egy lehetséges kimenete:

```
Név kombinációk program

Az első vezetéknév = Kiss
A második vezetéknév = Nagy
Az első keresztnév = Alma
A második keresztnév = Dávid

A lehetséges névkombinációk:
Kiss Alma
Kiss Dávid
Nagy Alma
Nagy Dávid
```

16. NÉV SZÉTBONTÁSA

Olassunk be egy nevet (egyetlen vezetéket- és keresztnévvel), majd válasszuk szét vezetéket- és keresztnévre. (Segítség: az első szóköz mentén kell szétvágni a szöveget a szóköz előtti, ill. utáni részekre.) A teljesNév változóban a szóköz karakter megkeresése: **teljesNév.IndexOf(' ')**, mely visszaadja a karakter sorszámát, ill. -1-et, ha nem szerepel a szövegben a karakter. A szövegből egy karaktersorozatot a **Substring** metódussal tudunk kivenni (ld. dokumentáció elején).

Ha a beírt névben nincs szóköz, akkor írja ki, hogy nem lehet kettészédeni, ha viszont 1-nél több szóköz van, akkor is csak az első szóköznél vágjuk szét a nevet.

A program egy kimenete lehet:

```
Név szétbontása program

A teljes név = Kiss Béla

A vezetéknév: Kiss
A keresztnév: Béla
```

17. KIS-NAGYBETŰ PÁR

Kérjünk be egy karaktert, ha nagybetű, írjuk ki a kisbetűs párját, ha kisbetű, akkor fordítva. (Egyéb karakterekre nem kell működnie). A kis- és nagybetű megállapításához, valamint a párja megkereséséhez használhatjuk a karakterek ASCII kódját, vagy a fordítókörnyezet kis- illetve nagybetűssé alakító függvényét. A programot úgy készítsük el, hogy ne csak az angol ABC betűire működjön, hanem az ékezetes karakterekre is!

Segítségképp néhány karakter ASCII kódja: A (65) - a (97), Z (90) - z (122) (figyeljük meg, hogy a különbség minden esetben 32, ezt ki tudjuk használni a karakter-pár megtalálásához).

A program egy kimenete lehet:

```
Kis-nagybetű pár program  
Mi lesz a karakter? é  
A beírt karakter kisbetű volt, párja: é - É
```

```
Kis-nagybetű pár program  
Mi lesz a karakter? r  
A beírt karakter kisbetű volt, párja: r - R
```

18. BETŰKERESŐ

Kérjünk be egy mondatot, majd egy karaktert. Írjuk ki, hogy az adott karakter szerepel-e a mondatban, s ha igen, akkor azt is, hogy hányszor (és hol = hányadik karakterként) szerepel a mondatban. A keresést az ismert programozási tételek segítségével kell megoldani.

A program kétféle keresést valósítson meg:

- a., a kis- és nagybetűket különbözőeknek tekintjük
- b., a kis- és nagybetűket nem tekintjük különbözőeknek

A program egy lehetséges kimenete:

```
Betűkereső program  
A mondat = Ebben a mondatban keresünk.  
A keresendő karakter = E  
Ha számít a kis- és nagybetű eltérés:  
A mondatban 1 db "E" karakter található. Helye a(z) 1. karakter.  
Ha nem számít a kis- és nagybetű eltérés:  
A mondatban 4 db "e" vagy "E" karakter található. Helye a(z) 1, 4, 20, 22. karakter.
```