3. feladatsor
Speciális szoftverek – LaTeX

Vadon Viktória

2023/24/I. félév

Szükséges csomagok

- subcaption
- array
- multirow
- xcolor vagy colortbl
- wrapfig
- float
- listings
- *szorgalmihoz*: longtable, algpseudocode, algorithm

Korábban már használt, de most is szükséges csomagok

- graphicx
- hulipsum, vagy lipsum, vagy blindtext
- hyperref?

1. Feladat (Ábrák)**.**

a) milyen csomagra lesz szükség képek beillesztéséhez? töltsük be!

b) illesszünk be egy képet max. 5cm-es szélességgel *és* max. 5cm-es magassággal, torzítás nélkül!

c) a kép maradjon úsztatás nélkül, a „szöveggel egy sorban". előtte és utána is generáljunk egy-két bekezdés zagyva szöveget. mi történik?

d) illesszünk be egy másik képet és helyezzük *úszó* ábra környezetbe. figyeljük meg, mi történik!
   - ha furcsa helyre kerül, próbálkozzunk különböző elhelyezési beállításokkal!

e) a korábbi úszó ábra környezetbe helyezzünk el pluszban egy másik, valamilyen módon transzformált (tükrözött, nyújtott, keretezett, stb.) képet!

f) adjunk feliratot ennek a transzformált képnek is. mi történik, ha beillesztünk egy második feliratot az ábrába?

g) a dokumentum elején listáztassuk a dokumentum ábráit!
   - a mappába, ahol a .tex fájlunk van, keressük és nézzük meg a .lof fájlt!

h) az ábra környezeten belül készítsünk *részábra* környezetet mindkét képnek, és rendezzük el őket egymás mellett.

i) feliratozzuk az ábrát és a részábrákat is!

j) szépítsük az elrendezést: igazítsunk középre, vízszintesen és függőlegesen is, és tegyünk a részábrák közé némi térközt.

k) helyezzünk el `\label`-öket az ábrában és a részábrákban is! melyikre használható `\ref`, `\subref`? mi a különbség?

2. Feladat (Táblázatok)**.**

a) készítsük el az 1. táblázatot!
   - az oszlopok legyenek: egy 30pt széles, egy balra, egy középre és egy jobbra igazított
   - készítsünk négy sort
   - töltsük fel a cellákat szöveggel – néhány üresen marad
   - készítsük el a rácsozást – figyeljünk a dupla ill. részleges rácsvonalakra!

tipp *a további feladatokhoz készítsünk egy másolatot az 1. táblázatról és töröljük ki a bal szélső oszlopát, azzal dolgozzunk tovább.*

b) készítsük el legalább az egyik színes táblázatot a 2. táblázatból!
   - 2a. táblázat: sorok váltakozó színezése – vízszintes rácsvonal (is) automatizálható
   - 2b. táblázat: oszlopok, cellák és rácsvonalak színezése, szöveg színe

c) a dokumentum elején listáztassuk a táblázatokat!

d) készítsük el a 3. táblázatot!
   - hogyan tudunk vízszintesen, függőlegesen, illetve mindkét irányban egyesíteni cellákat?

e) helyezzük egy olyan úszó környezetbe, hogy a *szöveget körbefuttathassuk körülötte*!

3. Feladat (Verbatim)**.**

a) inline verbatim használatával helyezzünk el egy mondatba elszórva egy-két `\LaTeX` parancsot!

b) verbatim környezetben szemléltessük egy (rövid) enumerate lista LaTeX kódját!

4. Feladat (Programkód 1)**.**

a) a `listings` csomag segítségével töltsük be a mellékelt Python kódrészletet!
   - olvashatjuk a külső fájlból, vagy be is másolhatjuk a LaTeX kódba

tipp az 1. Python kód minta/inspiráció a formázáshoz

b) állítsuk be a programnyelvet az automatikus formázáshoz
c) állítsunk be 2 szóköznyi tabulátorszélességet, a szóközök és tabulátorok legyen láthatatlanok!
d) számozzuk a kódsorokat a bal oldalon, pl. 4-esével
e) kísérletezzünk a keretezéssel, és válasszunk egy szimpatikus megjelenést!
   - ha szükséges, növeljük meg a bal margót, hogy a sorszámozás a kereten belülre kerüljön!

5. Feladat (Programkód 2)**.**

a) a `listings` csomag segítségével töltsük be *csak az egyik függvényt* a mellékelt C kódrészletből!
   - használjuk a külső fájlból való inputot!

tipp az 1. C kód minta/inspiráció a formázáshoz

b) a listings csomaggal definiáljunk egy *stílust* a kód színezésére
   - pl. legyen a háttér halványszürke, a kulcsszavak/parancsok sötétpirosak, a függvénynevek stb. sötétkékek, a kommentek pedig szürkék

c) alkalmazzuk ezt a stílust, illetve a C nyelvet a C kódunkra!
d) hogyan tudjuk elérni, hogy a tabulátorok és számozás formázása érvényben maradjon az előző kódrészlet után?
e) a keretet viszont távolítsuk el!

6. Feladat (Float)**.**

a) float csomag segítségével definiáljunk két új float környezetet, külön a Python és C kódoknak!
b) csomagoljuk a listing-jeinket a megfelelő float környezetekbe, és adjunk nekik caption-t!
   - definiáljuk a típus megjelenítendő nevét is, hogy a caption-ben ne a belső használatú típusnév jelenjen meg!

c) hogyan tudjuk elérni, hogy a caption a kódok fölött jelenlen meg?
d) készítsünk egy másolatot a C kódrészletről, és módosítsuk, hogy a másik függvényt tartalmazza! módosítsuk hozzá a caption-t is.
e) listáztassuk ki mindkét programtípust!

---

7. Szorgalmi feladat (Longtable)**.**

- kísérletezzünk a `longtable`-lel!
  - pl. ismételjük meg a `\caption`-t minden oldalon a *fejlécben*, tegyünk bele `\footnote`-ot, stb.

8. Szorgalmi feladat (Pszeudokód)**.**

a) ehhez a forrásanyag külön szorgalmi diasorban található!
b) töltsük be az `algpseudocode` csomagot!
c) készítsük el vele egy algoritmus pszeudokódját!

- pl. adatstruktúrák és algoritmusokból tanult; a dokumentum végén mellékelt a felosztás (2. algoritmus) és gyorsrendezés (1. algoritmus)

d) számozzuk a sorokat pl. kettesével!

e) töltsük be az `algorithm` csomagot, és csomagoljuk az `algorithm` környezetbe a pszeudokódunk, hogy úszhasson!

f) feliratozzuk és listáztassuk!
   - fordítsuk magyarra a lista és típus nevét!

g) `[haladó]` definiáljuk saját blokként a `do-while` ciklust! teszteljük is!
   - vigyázzunk, a parancsok ne egyezzenek meg létezőkkel!

h) `[haladó]` definiáljuk saját *folytatható* blokként a `switch-cases` esetszétválasztást!

1. táblázat. Igazítás, rácsvonalak

	egy	kettő	három
Helló világ!	négy	öt	hat
	hét	nyolc	kilenc
lórum ipse	tíz		tizenkettő

2. táblázat. Színezés

(a) Zebra

egy	kettő	három
négy	öt	hat
hét	nyolc	kilenc
tíz	tizenegy	tizenkettő

(b) Tarka

egy	kettő	három
négy	öt	hat
hét	nyolc	kilenc
tíz	tizenegy	tizenkettő

3. táblázat. Cellák összevonása

egy	kettő	
négy	öt	hat
tíz	nyolc	

1. Python program. Bináris keresés és beszúró rendezés Python-ban

```
def binary_search(arr, val, start, end):
  if start == end:
    if arr[start] > val:
      return start
    else:
      return start+1
  elif start > end:
    return start
  else:
    mid = (start+end)/2
    if arr[mid] < val:
      return binary_search(arr, val, mid+1, end)
    elif arr[mid] > val:
      return binary_search(arr, val, start, mid-1)
    else: # arr[mid] = val
      return mid

def insertion_sort(arr):
    for i in xrange(1, len(arr)):
        val = arr[i]
        j = binary_search(arr, val, 0, i-1)
        arr = arr[:j] + [val] + arr[j:i] + arr[i+1:]
    return arr
```

1. C program. Bináris keresés C-ben

```
binarySearch(arr, x, low, high)
  repeat till low = high
    mid = (low + high)/2
      if (x == arr[mid])
        return mid

      else if (x > arr[mid])  // x is on the right side
        low = mid + 1

      else       // x is on the left side
        high = mid - 1
```

1. Algoritmus Gyorsrendezés

```
    procedure QUICKSORT(@A,a,b)
Require: A írható tömb
Require: 1 ≤ a ≤ b ≤ Hossz[A] indexek
Ensure: a–b indextartományt rendezzük
 2:     if a=b then
            return A                          ▷ egyelemű tömb mindig rendezett
 4:     else
            FELOSZT(@A,a,b,A(a),@q)           ▷ k=A(a), a tartomány első eleme
 6:         QUICKSORT(@A,a,q)
            QUICKSORT(@A,q+1,b)
 8:         return A
        end if
10: end procedure
```

2. Algoritmus Felosztás

```
    procedure FELOSZT(@A,a,b,k,@q)
Require: A írható tömb
Require: 1 ≤ a ≤ b ≤ Hossz[A] indexek
Require: k A-beli kulcs
Ensure: A átrendezése és q választása úgy, hogy: a – q indextartomány elemei
    ≤ k, (q+1) – b indextartomány elemei ≥ k
 2:     i ← a-1
        j ← b+1
 4:     while i<j do                         ▷ ekvivalens: while true do...
            repeat
 6:             INC(i)
            until A(i) ≥ k
 8:         repeat
                DEC(j)
10:         until A(j) ≤ k
            if i < j then
12:             A(i) ↔ A(j) csere
            else
14:             q ← j
                return (A,q)
16:         end if
        end while
18: end procedure
```