

```

Deklaratív programozás pótzárthelyi, 2003. május 6.
=====
SML megoldások, 1.1 változat, dp03s-zh1-mlmegol-v1-1.txt
=====
-----
A csoport
-----
5. Mi az r típusa az alábbi, egymástól független deklarációk kiértékelése után?
(a) fun r (x, y) = (y, y x)      r : 'a * ('a -> 'b) -> ('a -> 'b) * 'b
(b) fun r x y = ((x y) y)       r : ('a -> 'a -> 'b) -> 'a -> 'b
(c) fun r x y = (x, x y)        r : ('a -> 'b) -> 'a -> ('a -> 'b) * 'b

Pontozás (összesen max. 6 pont):
5.a - 5.c Helyes válasz 2-2 pont.

6. Mi a v értéke és típusa az alábbi, egymástól független deklarációk kiértékelése után?
(a) val (v::w::_) = tl(explode "irat")      v = #"r" : char
(b) val (v, "v") = hd(List.filter (fn (x,y) => y="v") ["va", "vk", "v", "z"], ("5", "v"))]
(c) val v = implode (map (fn x => chr(ord x + 2)) ["#A", "#F", "#R"])      v = "CHT" : string

Pontozás (összesen max. 6 pont):
6.a - 6.c Helyes válasz 2-2 pont.

7. Az 'a drink adattípust így deklaráljuk:
datatype 'a drink = Water | Cola of 'a

(a) Írja fel Water, Cola és Cola "Light" típusát!      Water : 'a drink
                                   Cola : 'a -> 'a drink
                                   Cola "Light" : string drink

Tekintsük az f függvény definícióját:
fun f Water = 0
  | f (Cola "Light") = 199
  | f (Cola s) = if String.sub(s,0) < #"M" then 149 else 249

(b) Mi az x értéke az alábbi, egymástól független deklarációk kiértékelése után?
(b1) val x = f Water      x = 0
(b2) val x = f (Cola "Fanta")      x = 149
(b3) val x = f (Cola "Light")      x = 199
(b4) val x = f (Cola "Super")      x = 249

(c) Specifikálja az f függvény típusát!      f : string drink -> int

Pontozás (összesen max. 9 pont):
7.a Minden helyes válaszért 1-1 pont.
7.b1 és 7.b3 Helyes válaszért 1-1 pont.

```

```

7.b2 és 7.b4 Két helyes választért 3 pont, kis hibáért 1 pont, nagy hibáért 2 pont levonás.
7.c Helyes választért 1 pont.

8. Írjon SML-függvényt torlodok néven, amely megszámlálja, hogy egy magyar szóban hány helyen van mássalhangzótorlódás. Mássalhangzótorlódásnak nevezzük a három szomszédos mássalhangzóból álló betűcsoportot, amelyben az egymás mellett állók betűk nem azonosak. A többjegyű mássalhangzók (pl. a cs és a gy) minden betűjét külön mássalhangzónak tekintjük. A vizsgálandó szavakban csak betűk fordulnak elő. Nem kell hatékony megoldásra törekednie. Segédfüggvényt definiálhat (fejkommenttel!).

Használhatja a massal="bBcCdDfFgGhHjJkKlLmMnNpPqQrRsStTvVwWxXyYzZ" értéket, továbbá az isMem : 'a * 'a list -> bool tesztelő függvényt annak vizsgálatára, hogy egy adott érték benne van-e egy adott listában.

(* torlodok : string -> int
   torlodok s = az s szóban lévő hármas mássalhangzótorlódások száma
*)

Példák: torlodok "vegyszerboltban" = 3;
         torlodok "gyümölcstorta" = 2;
         torlodok "gyümölcszörppel" = 1;
         torlodok "kettoshangzo" = 1;
         torlodok "pattogatott" = 0;
         torlodok "" = 0;

Egy megvalósítása:

(* isMem : 'a * 'a list -> bool
   isMem (x, zs) = igaz, ha x benne van zs-ben
*)
fun isMem (x, []) = false
  | isMem (x, z::zs) = x=z orelse isMem(x, zs)

val massal = "bBcCdDfFgGhHjJkKlLmMnNpPqQrRsStTvVwWxXyYzZ"

fun torlodok s =
let
  val mas = explode massal
  (* tor : int list * int -> int
   tor (zs, n) = n + a zs listával ábrázolt szóban lévő
   hármas mássalhangzótorlódások száma
   *)
  fun tor (x::y::z::zs, n) =
    if x<>y andalso y<>z andalso isMem(x, mas) andalso
       isMem(y, mas) andalso isMem(z, mas)
    then tor(y::z::zs, n+1)
    else tor(y::z::zs, n)
  | tor (_, n) = n
in
  tor (explode s, 0)
end

Pontozás (összesen max. 9 pont):
Nem követelmény a hatékony program.

```

```

-----
B csoport
-----
5. Mi a t típusa az alábbi, egymástól független deklarációk kiértékelése után?
(a) fun t (x, y) = (x, (x, y)) t : 'a * 'b -> 'a * ('a * 'b)
(b) fun t x y = (x y, x y y) t : ('a->'a->'b) -> 'a -> ('a->'b) * 'b
(c) fun t y x = ((y x) x) t : ('a->'a->'b) -> 'a -> 'b

```

Pontozás (összesen max. 6 pont):

5.a - 5.c Helyes válasz 2-2 pont.

6. Mi a v értéke és típusa az alábbi, egymástól független deklarációk kiértékelése után?

```

(a) val ((_::::):v):_ = [explode "irat"] v = [{"a", #"t"}] : char list
(b) val ("v", v) = hd(List.filter (fn (x,y) => x="v") [{"va", "vk"}, {"v", "z"}, {"5", "v"}]) v = "z" : string
(c) val v = implode (map (fn x => chr(ord x - 2)) [{"C", #"H", #"S"}]) v = "AFQ" : string

```

Pontozás (összesen max. 6 pont):

6.a - 6.c Helyes válasz 2-2 pont.

7. Az 'a food adattípust így deklaráljuk:  
datatype 'a food = Air | Meat of 'a

```

(a) Írja fel Air, Meat és Meat "Frog" típusát!
Air : 'a food
Meat : 'a -> 'a food
Meat "Frog" : string food

```

Tekintsük az f függvény definícióját:

```

fun f Air = 0
  | f (Meat "Frog") = 219
  | f (Meat t) = if String.sub(t,0) > #"M" then 249 else 189

```

(b) Mi az x értéke az alábbi, egymástól független deklarációk kiértékelése után?

```

(b1) val x = f Air val x = 0
(b2) val x = f (Meat "Chicken") val x = 189
(b3) val x = f (Meat "Frog") val x = 219
(b4) val x = f (Meat "Rind") val x = 249

```

(c) Specifikálja az f függvény típusát! f : string food -> int

Pontozás (összesen max. 9 pont):

7.a Minden helyes válaszért 1-1 pont.

7.b1 és 7.b3 Helyes válaszért 1-1 pont.

7.b2 és 7.b4 Két helyes válaszért 3 pont, kis hibáért 1 pont, nagy hibáért 2 pont levonás.

7.c Helyes válaszért 1 pont.

8. Írjon SML-függvényt hangrend néven, amely meghatározza, hogy egy magyar szó magas, mély vagy vegyes hangrendű-e. A függvény eredménye

legyen l, ha az argumentuma magas, l, ha mély, és 0, ha vegyes hangrendű szó. Egy magas hangrendű szóban vannak magas hangrendű magánhangzók (e, é, ő, o, ü, u), de nincsenek mély hangrendűek (a, á, o, ó, u, ú). Egy mély hangrendű szóban vannak mély hangrendű magánhangzók, de nincsenek magas hangrendűek. Minden más szó (az üres szó is!) vegyes hangrendű. Az l és i magánhangzók a hangrend szempontjából közömbösek. Nem kell hatékony megoldásra törekednie. Segédfüggvényt definiálhat (fejkommenttel!).

Haszználhatja a magas="eEéÉőÖüÜűű" és mely="aÁáÁoOóóúÚú" értékeket, továbbá az isMem : 'a \* 'a list -> bool tesztelő függvényt annak vizsgálatára, hogy egy adott érték benne van-e egy adott listában.

```

(* hangrend : string -> int
   hangrend s = l, ha magas, ~l, ha mély, 0, ha vegyes hangrendű szó
   az s
*)

```

```

Példák: hangrend "örömszülő" = l;
         hangrend "kitekintő" = l;
         hangrend "ormótlan" = ~l;
         hangrend "viharos" = ~l;
         hangrend "méregpohár" = 0;
         hangrend "licit" = 0;
         hangrend "psszt" = 0;
         hangrend "" = 0;

```

Egy megvalósítása:

```

(* isMem : 'a * 'a list -> bool
   isMem (x, zs) = igaz, ha x benne van zs-ben
*)
fun isMem (x, []) = false
  | isMem (x, z::zs) = x=z orelse isMem(x, zs)

val magas = "eEéÉőÖüÜűű"
val mely = "aÁáÁoOóóúÚú"

fun hangrend t =
  let val mgs = explode magas
      val mly = explode mely
      (* hr : int list -> int
         hr xs r = l, ha magas, ~l, ha mély, 0, ha vegyes hangrendű
         *)
      fun hr (x::xs) (~l) = if isMem(x, mgs) then 0 else hr xs (~l)
        | hr (x::xs) l = if isMem(x, mly) then 0 else hr xs l
        | hr (x::xs) r = hr xs (if isMem(x, mly) then ~l
                                else if isMem(x, mgs) then 1
                                else r)
      in
        | hr [] r = r
      in
        hr (explode t) 0
      end

```

Pontozás (összesen max. 9 pont):

Nem követelmény a hatékony program.