

# 1. Vzdálená správa kontaktů pomocí WWW stránek

*Semestrálka určena pro dvě dvojice.*

Vytvořte aplikaci typu server-client, která umožní spravovat informace o vašich kontaktech. Info bude přístupné přes CGI skript vašich www stránek. Práci lze rozdělit na dvě části:

- Server-Client komunikace
- CGI skript na zobrazování stránek

Informace přístupná přes Web bude obsahovat adresář osob. Údaje se budou zapisovat do jednoho textového souboru jako samostatné položky. Každá položka musí obsahovat nejméně:

- Jméno a příjmení
- Adresu
- Telefon
- E-mail

Syntaxe oddělování položek v souboru a uvnitř jednotlivých položek je volná, určete si nejvhodnější. Minimalní požadavek na syntaxi je následující:

```
{           #zde musí být jakýkoliv oddělovač položek
jmeno = .....,
adresa = .....,
telefon = hodnota, # oddelovače jsou libovolné, toto je minimalní požadavek
...
}           #zde musí být jakýkoliv oddělovač položek
```

Jednotlivé položky budou načítány nejprve jako bloky a poté se z nich bude získávat potřebná informace. Zpracování dat bude prováděno pomocí regulárních výrazů.

## CGI Skript

Po zavolání skriptu se uživateli zobrazí stránka s informační tabulkou, která bude ve sloupcích obsahovat jednotlivá informační pole a v řádcích všechny kontakty. Někakým způsobem (třeba pomocí checkboxů) bude možné upravit strukturu tabulky, tedy určit, která pole se budou zobrazovat. Kliknutím na emailovou adresu v tabulce bude možné dotyčnému poslat email.

## Server-client část

Jako autor systému (a správce kontaktů) budete mít možnost aktualizovat informace v souboru, a to z jakéhokoliv počítače, pomocí připojení typu telnet jako klient k Vašemu serveru. (Připojením typu telnet se myslí vlastní klient se znakovým rozhraním komunikující se serverem přes TCP/IP, nikoliv CGI skript).

Po přihlášení proběhne jednoduchá autorizace (třeba jedním, pevně zvoleným heslem, nebo jménem autora). Přihlášený uživatel bude mít v podstatě dvě možnosti:

1. vytvořit nový záznam
2. smazat již existující záznam

Volba typu operace bude probíhat po přihlášení uživatele a po každém ukončení předcházející operace. Mazání již existujících záznamů provádějte podle jména kontaktu. Zbytek komunikace a syntax příkazů si zvolte libovolně.

Nezapomeňte ošetřit otevírání souboru na zápis! Do databáze se může najednou přihlásit více uživatelů a pouze jeden bude oprávněn provádět změny!

## 2. Statistický průzkum pomocí WWW stránek

*Semestrálka určena pro jednu dvojici.*

Vytvořte CGI skript, který bude provádět jednoduchý statistický průzkum čehokoli (sledovanosti TV programů, souhlasu s útoky na Afghánistán, atd.).

Vyvoláním stránky s CGI skriptem se zobrazí výsledky právě probíhajícího průzkumu, tzn. např. nadpis, otázka a dosavadní výsledky ve formě histogramu s udáním procentuálních i absolutních čísel (histogram může mít libovolný formát – stojatý i ležatý, nemusí být vyloženě grafický). Pokud zrovna nebude žádný průzkum probíhat, zobrazí se místo grafů jen hlášení. Pod histogramem bude dotazník s možnými odpověďmi (s checkboxy), které umožní návštěvníkovi stránky hlasovat. Poslední na stránce bude tlačítko pro správce systému.

Po kliknutí na toto tlačítko se jednoduše ověří, zda se opravdu jedná o správce (např. porovnáním s jedním, předem známým heslem). Následně se zobrazí stránka pro definici statistického průzkumu. Ta bude obsahovat:

- kolonku pro pokládanou otázku (která se následně bude zobrazovat nad sloupcovými grafy a měla by vyjadřovat cíl výzkumu),
- kolonku pro možné odpovědi (všechny se zadají do jediného řetězce, odliší se pomocí oddělovačů).

Tato definice průzkumu bude uložena do textového souboru. Formát souboru je libovolný, doporučuje se obdoba následujícího:

```
{  
  otazka = ...;  
  pocetmoznosti = ...;  
  odpoved1 = ...;  
  odpoved2 = ...;  
  ...  
  cetnost_odp1 = hodnota;  
  cetnost_odp2 = hodnota;  
  ...  
}
```

Zpracování textů bude probíhat pomocí regulárních výrazů.

### 3. Bankovní konto

*Semestrálka určena pro jednu dvojici.*

Vytvořte pomocí CGI skriptu simulaci vašeho bankovního konta. Konto bude podporovat následující operace:

Pro všechny:

- přihlášení a autorizace účastníka transakce

Pro uživatele konta:

- přehled o stavu konta
- převod peněz na jiné konto
- vklad peněz
- výběr peněz v hotovosti

Pro všemocného správce banky:

- výpis informací o všech kontech (uživatel, heslo, stav konta)
- zavedení konta
- zrušení konta

Převod peněz bude možný pouze v rámci této banky na již existující konto. Nepodporujeme konta s možností úvěru, zůstatek musí být kladný.

Informace o kontech ukládejte do jednoho textového souboru. Formát oddělovačů je libovolný. Minimální dělení do skupin bude následující:

```
{  
cislokonta = ...,  
jmeno = ...,  
heslo = ...,  
zustatek = ...,  
kdyzalozeno = ...,  
}
```

## 4. Hra „Patnáctka“

*Semestrálka určena pro jednu dvojici.*

Vytvořte CGI skript, který umožní zahrát si hru „15“. Je to klasická šoupací dlaždicová skládačka. V poli 4x4 je umístěno 15 očíslovaných dlaždic, jedno pole je volné. Toto volné pole lze využít pomocí přešoupávání dlaždic k jejich přeskupení do cílového stavu, kdy bude 1. řádek obsahovat dlaždice 1-4, 2. řádek 5-8, atd.

Po zavolání skriptu se objeví tabulka (formát libovolný, možno udělat graficky i textově) představující rozložení dlaždic v hrací mřížce. Pod tabulkou bude minimálně 7 tlačítek: 4 pro posun dlaždice nahoru, dolů, doleva nebo doprava (pokaždé se jedná o jinou dlaždici), tlačítko Restart, které vymaže historii tahů a náhodně rozhází dlaždice, tlačítko High-scores, které zobrazí tabulku s 10 nejlepšími dosaženými výsledky, a tlačítko Undo, které umožní vzít zpět poslední z historie tahů. (Ovládání je možné upravit do jiného stylu, např. přepínač – radiobutton, kterým nastavíte požadovanou akci a jedním tlačítkem typu Submit info o požadované akci odešlete CGI skriptu. Upravené ovládání ale musí respektovat požadované funkce.)

Skript musí kontrolovat proveditelnost tahu (na neproveditelné tahy buď nebude reagovat, zobrazí upozornění nebo přímo tlačítka, která by neproveditelný tah mohla způsobit, předem zakáže).

Bude se udržovat úplná historie tahů tak, aby se opakovaným stiskem tlačítka Undo dalo dojít až do počáteční konfigurace hry.

Výpis nejlepších výsledků bude obsahovat jméno hráče a dosažený počet tahů (čím méně, tím lépe, samozřejmě). K zadání jména bude hráč vyzván po ukončení hry, pokud jeho skóre bude patřit mezi 10 nejlepších.

**POZOR:** Hra se musí být řešitelná! Tzn. musí existovat způsob, jak dlaždice uspořádat do cílového stavu. To lze docílit dvěma způsoby:

Počáteční rozvržení dlaždic generovat náhodně a kontrolovat řešitelnost, nebo (což je jednodušší) počáteční rozvržení tvořit z cílového stavu a přípustnými tahy tuto konfiguraci rozházet.

Které řešení zvolíte je už na vás.

## 5. Hra „NIM“

*Semestrálka určena pro jednu až tři dvojice*

Vytvořte CGI skript, který umožní zahrát si hru „NIM“. Hra je určena pro dva hráče, kteří se postupně střídají v tazích. Hrací plocha obsahuje 4 řady sirek (náhodné počty, řekněme v intervalu 20 - 40). Každý hráč může odebrat 1 až 3 sirky, vždy ale jen z jediné řady sirek. Ten, který musí odebrat z hrací plochy poslední sirku, prohrává.

Program lze rozdělit na dvě části:

- CGI skript rozhraní
- Skript strategie

### **CGI skript rozhraní (1 dvojice)**

Hráči (lidé nebo počítačové strategie) se budou střídat v tazích. Skript bude libovolným způsobem zobrazovat stav hry (ale alespoň trochu grafickým – nezobrazovat jen počty sirek, ale třeba každou sirku reprezentovat i znakem I). Pod zobrazením bude malý formulář pro hráče, který je právě na tahu, jímž se bude dát určit typ hráče (člověk, počítač), číslo řady sirek, z níž bude chtít odebírat, a počet odebíraných sirek. Tvar formuláře bude libovolný (edit boxy, spinboxy, ...). Tahy se budou ukládat do souboru; formát si zvolte sami.

Skript musí umožňovat (nikoli realizovat) hru počítače proti člověku, příp. počítače proti počítači. Tah prováděný počítačem bude realizovat skript strategie, který není součástí této části úlohy. CGI skript musí předem vědět, zda nějaký skript strategie existuje a jeho jméno. Bude od něj očekávat, že jeho zavoláním přibude do souboru s uloženými tahy jeden záznam vyjadřující tah počítače. CGI skript tedy v tomto případě prostě spustí skript strategie, následně přečte poslední řádek ve sdíleném souboru a dál pokračuje, jako by tyto údaje získal od člověka, tedy vytvoří HTML stránku, která bude zobrazovat aktuální stav hry.

### **Skript strategie (1 až 2 dvojice)**

Vše, co bude mít za úkol tento skript, je pouze přečíst si obsah sdíleného souboru se záznamem historie tahů, a tímto způsobem zjistit aktuální stav hry. Z těchto informací musí vydedukovat další nejvhodnější tah a tento tah zapsat zpět do sdíleného souboru se záznamem tahů (pokud se skupiny dohodnou na jiném způsobu předávání informací, je to možné provádět jinak).

Strategie by měla být alespoň trochu inteligentní, tzn. neměla by jen náhodně volit řadu a počet sirek. Pokud budou strategie vytvářet dvě skupiny, bylo by dobré, aby pracovali odděleně a aby vyvinuly dvě odlišné strategie, které by bylo možné nechat hrát vzájemně proti sobě.

### **Pojmenování a formát souborů**

Jelikož v této úloze bude docházet ke sdílení některých informací, doporučuji (pokud se skupiny nedohodnou jinak), aby se soubor se záznamem tahů jednotlivých hráčů jmenoval *tahy.dat* a aby měl textový formát. Každý řádek v tomto souboru bude reprezentovat 1 tah a např. řetězec „2 15 5 17 0“ bude záznamem situace, kdy po tahu hráče č.2 zbylo v prvním řádku 15 sirek, v druhém 5, ve třetím 17 a ve čtvrtém 0. Skripty strategií doporučuji pojmenovat jako *strat1.pl*, příp. *strat2.pl*.

## 6. Sledování přístupů na stránky

*Pro jednu dvojici*

Vytvořte CGI skript, který bude realizovat rozšířené počítadlo přístupů na stránky. Pro jednotlivé uživatele bude počítadlo udržovat statistiky o přístupech, tzn. v jaký den se uživatel přihlásil, v kolik hodin se přihlásil, v kolik hodin se odhlásil, atd. Správci počítadla (uživatel se zvoleným jménem, např. „root“) skript umožní vypsat info o všech uživatelích, ostatním uživatelům jen info o nich samotných.

Po spuštění CGI skriptu se zobrazí výzva pro zadání uživatelského jména. Následně se zkontroluje, zda toto jméno již v systému existuje. Pokud ano, záznamy budou doplňovány k existujícímu uživateli, pokud ne, automaticky se uživatel tohoto jména založí. Následně CGI skript vygeneruje dynamickou HTML stránku, která bude rozdělena na dva rámce. V prvním, úzkém rámci umístěném buď při levém nebo při horním okraji okna prohlížeče se bude zobrazovat tlačítkové menu CGI skriptu. Bude obsahovat tlačítka

- pro odhlášení,
- pro zobrazení statistik,
- pro zobrazení zvolené statické stránky a
- bude libovolným způsobem indikovat, kolikráté přihlášení uživatele k CGI skriptu to je.

Zvolenou statickou stránkou se myslí jakákoliv HTML stránka (ať už vámi vytvořená, nebo již existující někde na internetu, např. seznam.cz), která bude dopředu vámi vybrána. Není nutné, aby se dala pomocí CGI skriptu měnit (aktivitě se meze nekladou).

V druhém rámci se tedy bude zobrazovat buď zvolená statická stránka nebo HTML stránka generovaná CGI skriptem obsahující statistiky o přístupech přihlášeného uživatele (nebo o přístupech všech uživatelů v případě uživatele „root“). Tyto statistiky by měly obsahovat počet přístupů, průměrnou dobu trvání jednoho přístupu, celkovou dobu trvání přístupů a tabulku, v níž se zobrazí kompletně všechny záznamy o všech přístupech přihlášeného uživatele. V případě správce (root) bude existovat možnost zobrazit info o všech uživatelích. (Hlavně v případě uživatele „root“ by bylo vhodné, aby se záznamy daly řadit podle jména uživatele, času přihlášení, apod., ale toto není podmínkou pro úspěšné splnění úlohy.)

Způsob uložení údajů je libovolný a řešitelé úlohy si jej mohou jakkoli přizpůsobit. Mohou odděleně udržovat soubor se záznamy o přihlášení a soubor se jmény uživatelů (u nichž si mohou ukládat „předpočítané“ celkové statistiky), nebo mohou udržovat jen jeden soubor, kde bude mít např. každý řádek formát:

uživatel, přihlášení/odhlášení, datum, čas

a celkové statistiky budou počítat až „za běhu“. Konečné řešení je zcela v rukou řešitelů.

## 7. Webová kalkulačka

*Semestrálka pro 1 dvojici*

Vytvořte CGI skript, který bude realizovat jednoduchou kalkulačku pomocí HTML stránky.

CGI skript bude zobrazovat kalkulačku ve formě formuláře, který bude obsahovat pole s vypočítaným mezivýsledkem, textové pole, kam se bude zadávat operand vstupující společně s mezivýsledkem do následující operace. Typ operace se bude zadávat pomocí přepínače (radio button), jehož jednotlivé položky budou obsahovat:

- + sčítání
- - odčítání
- \* násobení
- / dělení
- = výpočet konečného výsledku
- C vymazání mezivýsledku
- Hist výpis historie kalkulačky
- Smaz smazání historie kalkulačky

Formulář musí ještě obsahovat tlačítko typu Submit, jímž se informace budou odesílat CGI skriptu.

Základní matematické funkce (+-\*/), funkce pro výpočet a vymazání a zobrazení historie jsou minimálním požadavkem (pokud chcete doplnit více funkcí, nebude to na závadu). Skript bude udržovat v nějakém souboru záznamy o provedených výpočtech. Akce, které vyvolávají jednotlivá tlačítka, jsou zřejmé. Tlačítko history způsobí vygenerování stránky, na níž budou vhodnou formou zobrazeny provedené výpočty (forma jejich zobrazení zůstává na řešitelích, musí však být zřetelná v tom smyslu, aby se z ní daly snadno rozeznat jednotlivé operace). Stránka s historií musí obsahovat možnost, jak se vrátit zpět na kalkulačku.

Je na řešitelích, zda bude kalkulačka respektovat prioritu operátorů, tj. zda bude upřednostňovat násobení a dělení před sčítáním a odčítáním, či zda bude mezivýsledky a operátory brát tak, jak „leží a běží“. Ale POZOR! Zvolenému způsobu zpracování musí také odpovídat zápis provedeného výpočtu ve výpisu historie!

Příklad: Předpokládejme, že uživatel provede následující operace:

5,+2,\*3,=

V případě, že se prioritu rozhodnete respektovat, by měl být výsledek 11 a řádek v souboru s historií může vypadat opravdu jen takhle:

5+2\*3=11.

V případě, že prioritu respektovat nebudete, výsledek by měl být 21 a řádek v souboru s historií musí vypadat odlišně od minulého případu.

(5+2)\*3=21

Za nesprávné řešení „závorkování“ se v tomto případě považuje i nadbytečnost závorek, tzn. závorkování každé operace, které by vedlo na zápis

(((a+b)\*c)/d)-e=x

## 8. Slovník a výuka slovíček

*Semestrálka pro 1 dvojici*

Vytvořte CGI skript, který bude umožňovat výuku slovíček (předpokládejme angličtinu) přes webové rozhraní.

Zadáním adresy CGI skriptu v okně prohlížeče se uživatel dostane na stránku, která ho vyzve k zadání uživatelského jména. Pokud uživatel s tímto jménem neexistuje, bude založen.

*(Předpokládejme, že se systémem budou pracovat jen rozumní uživatelé, kteří ho nebudou poškozovat a budou ho využívat nejen ku prospěchu svému, ale také ku prospěchu ostatních. Nerozumným chováním se myslí např. to, že by uživatel schválně do systému vkládal špatné nebo nesmyslné překlady.)*

Každý přihlášený uživatel bude mít potom následující možnosti:

- Použít systém jako slovník
- Použít systém pro zkoušení slovíček
- Odhlásit se

Slovíčka budou uložena v textovém souboru, který bude mít např. následující formát:

```
--- # Oddělovač záznamů
cz = dum;
eng = house;
owner = all;
---
```

Význam polí *cz* a *eng* je zřejmý. Pokud jednomu českému výrazu odpovídá více výrazů anglických (nebo opačně), je třeba každou dvojici zapsat jako zvláštní záznam. Pole *owner* slouží k identifikaci, komu má být dané slovíčko přístupné při zkoušení (viz dále).

### Slovník

Tato část bude umožňovat jednak slovíčka zadávat a jednak je vyhledávat.

Při zadávání bude mít uživatel k dispozici formulář se dvěma edit-boxy (pro český a anglický výraz) a třeba přepínač (radio-button) s určením, zda je slovíčko určeno jen pro právě přihlášeného uživatele nebo pro všechny. Cílem je odlišit výrazy, které se týkají speciálně oblasti zájmu uživatele (všechny asi nebude zajímat, jak se anglicky řekne třeba „nábojová pumpa“) od obecných výrazů (naopak asi všichni by měli vědět, jak se řekne třeba „dům“).

Při vyhledávání ve slovníku by měl uživatel dostat možnost určit, zda chce provádět překlad čeština->angličtina, nebo opačně, a zadat vyhledávaný výraz. Skript by měl poskytnout stránku se seznamem všech odpovídajících výrazů v druhém jazyce (ať už je jejich vlastníkem kdokoli).

### Zkoušení

Uživatel by měl opět mít možnost určit, jestli má být zkoušen ve směru anglicko-českém nebo česko-anglickém. Zkoušení by mělo probíhat jen ze slovíček, které patří všem uživatelům a právě přihlášenému uživateli. Ostatní by se ve zkoušení objevovat neměla.

Pravděpodobnost výběru každého konkrétního výrazu by se měla měnit podle úspěšnosti uživatele při jeho překladu. Když slovíčko přeloží správně, mělo by se dále vybírat s menší pravděpodobností, když slovíčko přeloží špatně, měla by se pravděpodobnost zvýšit. Na počátku zkoušení mají všechna relevantní slovíčka stejnou pravděpodobnost výběru.

Současně s formulářem pro zkoušení by se měly zobrazovat jednoduché statistiky o počtu správných a špatných odpovědí. Úspěšnost při zkoušení a pravděpodobnosti výběru slov udržujte pro jednoduchost jen v rámci jednoho sezení (od přihlášení po odhlášení). Předpokládejte, že všichni uživatelé jsou poslušní a že se ze systému budou odhlašovat.



## 9. Seznam úkolů

*Semestrálka pro 1 dvojici*

Vytvořte CGI skript, který bude generovat webovou stránku se seznamem úkolů.

Představme si firmičku, kde jsou si všichni rovni a kde může každý každému zadávat úkoly. Zadáním adresy skriptu v prohlížeči se zobrazí stránka se všemi zadanými úkoly přehledně uspořádanými v tabulce.

O každém úkolu by měly být známy následující informace:

- Popis úkolu
- Kdo úkol zadal
- Komu byl zadán
- Jakou má prioritu (např. 1 nejnižší – 5 nejvyšší)
- Kdy byl úkol zadán
- Do kdy má být vyřešen
- Kdy skutečně byl vyřešen

Jména zaměstnanců firmy budou předem známa a budou uložena v textovém souboru (např. *zamestnanci.txt*). Uživatel systému (zaměstnanec) se bude muset přihlásit (svým jménem), aby mohl v systému dělat něco víc. Po přihlášení bude mít následující možnosti:

- Prohlédnout si všechny jím zadané úkoly
- Zadat nový úkol
- Zrušit některé jím zadané úkoly
- Prohlédnout si všechny úkoly, které má vyřešit
- Označit některé z těchto úkolů za hotové

Uživatel samozřejmě může úkol zadat sám sobě.

Uživatel by měl také mít možnost zvolit si rozsah zobrazovaných informací o úkolech. Předpokládejme, že se seznam úkolů bude zobrazovat v tabulce – co řádek, to jeden úkol. Pro každou informaci o úkolu (popis, kdo, komu, kdy, ...) by se měl ve formuláři objevit check-box, který umožní určit, zda se příslušný sloupec v tabulce má nebo nemá zobrazovat.

Zadání nového úkolu by mělo být realizováno samostatným formulářem. Rušení úkolů a označování za hotové je možné realizovat tak, že se v tabulce ke každému úkolu přiřadí check-box, jejichž zaškrtnutím se určí se kterými záznamy chceme pracovat. Pomocí přepínače (umístěného třeba pod tabulkou) se pak jen určí, co se má s označenými záznamy provést (zrušit, označit za hotové).

Dejte pozor na to, aby rušit úkoly mohly jen ty osoby, které úkol zadaly. Stejně tak může být úkol označen za hotový jen tou osobou, pro kterou byl úkol zadán.

## 10. Správa souborů pomocí WWW stránek

*Semestrálka určená pro 1 dvojici*

Vytvořte CGI skript, který zobrazí jako WWW stránku soubory v aktuálním adresáři (aktuální adresář si libovolně zvolte) a také názvy podadresářů. Kliknutím na soubor si ho můžete stáhnout, kliknutím na podadresář se rozbalí struktura podadresáře.

Všechny soubory a adresáře budou mít vlevo od sebe checkbox. Zaškrtnutím checkboxu a odesláním informace CGI skriptu se adresář/soubor smaže. Pokud adresář není prázdný, bude následovat potvrzení pro smazání a po kladném potvrzení se rekurzivně smaže všechno v tomto adresáři.

Uživatel musí mít možnost vytvořit prázdný soubor a adresář. Vhodná metoda je udělat na stránce jeden neexistující adresář a soubor se jménem "nový adr/soubor" a po kliknutí na tuto položku se dotázat na jméno nového adresáře/souboru. Konečné řešení je na autorech práce.

## **11.Rodokmen**

*Semestrálka určená pro 1 dvojici*

Vytvořte CGI skript, který Vám umožní spravovat Váš rodokmen. Rodokmen se bude zobrazovat jako strom v textové podobě na WWW stránce. Do každé větve stromu bude možno přidat novou položku, která může být novým uzlem, nebo jen zatím posledním členem. Prostor pro vkládání nových položek můžete vyjádřit např. prázdným editačním boxem na konci každé větve stromu.

Informace o stromu se budou ukládat v textové formě do souboru. Po nahrání stránky se objeví aktuální verze stromu s možností přidávat další informace ke stromu.

Formát textového souboru si zvolte sami.

## **12.Dotazník předmětu OSA**

*Semestrálka určená pro 1 dvojici*

Vytvořte CGI skript s formuláře, podle dotazníku první hodiny. Obsah odpovědí ukládejte do vhodně formátovaného textového souboru. Ke každé otázce bude k dispozici statistika se všemi odpověďmi. Jména osob však nebudou přes webové rozhraní dostupné bez autentizace.

Pro přístup k celému souboru dat bude nutná jednotvářná autentizace, po níž bude možné zobrazit ke každému záznamu jméno i celý seznam odpovědí.

Zda umožníte odpovídat anonymně je na Vás. Pokud to umožníte nezapomeňte se vhodným způsobem vypořádat se zaznamenáváním anonymních odpovědí.

## **13.Správa zadání semestrálních prací pro OSA**

*Semestrálka pro jednotlivce*

Vytvořte webové rozhraní pomocí kterého se studenti budou moci hlásit o zadání jednotlivých semestrálních prací.

Ke každému zadání se bude moci zapsat pouze zadáním omezený počet lidí. Každý zapsaný obdrží po zapsání textový řetězec, který když zadá může se ze zadání opět smazat. Možnost zapisovat se a mazat bude omezena datem.

Jako vstupy a výstupy používejte pouze textové soubory.

Pokud to dokážete, vybavte skript posíláním emailové zprávy na adresu učitele, který ma cvičení na starost. Nezapomeňte, že jeden učitel může mít vícero hodin cvičení a je proto nutné rozlišovat přihlášení též podle cvičení.