Kryptografie

1 Statický web

Vytvořte webovou stránku, která bude reprezentovat vámi zvolené tři šifry. Může se jednat o velmi jednoduché šifry jako je například substituční šifra „Caesar“, nebo transpoziční šifra „Slova pozpátku“ atp. Abecedu si můžete zjednodušit na velké znaky A-Z. Ke každé šifře uveďte její název, popis, popis postupu šifrování/dešifrování, zdrojový kód. Na webu bude také možnost online zašifrovat/dešifrovat návštěvníkem zadaný text do textového pole. Navrhněte rozložení jednotlivých prvků webu. K realizaci použijte HTML, CSS a JavaScript. Webová stránka bude moderní, responzivní, validní, přístupná, SEO optimalizovaná, performance optimalizovaná a bude mít edukační charakter.

2 Databáze

Navrhněte databázi, ve které se budou evidovat nalezené zašifrované texty. Tento entitní typ budeme evidovat jako „message“ a u každého bude patrné co (obsah šifrované zprávy), kdy (datum a čas) a kde (GPS souřadnice) se nalezl a budou nést nějaké desetiznakové označení (např.: 2024-A-001). Tyto texty se budou kryptoanalytici pokoušet dešifrovat a každý takovýto pokus o dešifrování se bude zaznamenávat do databáze. O pokusu se bude evidovat, kdo jej provedl, kdy jej provedl, jak dlouho prací kryptoanalytik strávil, popis práce kryptoanalytika a informace, povedlo-li se šifrovaný text nakonec tímto pokusem rozšifrovat.
Databázi navrhněte v dat.md jako tabulky entitních typů. Sloupce tabulky: název atributu, datový typ, klíče, modifikátory integritního omezení.
K výše uvedené databázi připravte SQL dotazy, které:

1) Vytvoří tabulku pro šifrované zprávy s názvem „message“, která bude obsahovat nalezené šifry dle výše uvedeného zadání.
2) Uložení nově nalezeného zašifrovaného textu do DB. Tento text zní „joha“, nalezen byl zde (50.91295N, 14.6171E) a právě teď (v době zadání práce).
3) Smazání všech pokusů o rozšifrování zprávy s kódem 2024-B-003.
4) Seznam všech zašifrovaných zpráv (kód, datum – den nalezení) seřazených dle celkové délky pokusů o rozšifrování od nejdelšího. Tedy jak dlouho na nich kryptoanalytici strávili čas od toho nejnáročnějšího.
5) Seznam všech kryptoanalytiků (jméno) s počtem zpráv, které úspěšně rozluštili.
6) Pomocí SQL zajistěte, aby se po smazání záznamu z tabulky „message“ smazaly automaticky i pokusy o rozšifrování.

3 Program

Mějme problém šifrování pomocí šifrovací mřížky. Jedná se o transpoziční šifru, kde se pozice písmen v šifrovaném textu určuje dle otvorů ve čtvercové mřížce.
Mějme například miniaturní mřížku 2x2, která má otvor v pravém horním rohu. Zapsat ji můžeme například jako matici $\begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix}$. Nula značí otvor, do kterého můžeme psát. Písmena bereme z nešifrovaného textu postupně zleva. Vyčerpáme-li všechny otvory mřížky, kam vstupní písmena zapisujeme také zleva postupně po řádcích dolů, tak mřížku o 90° otočíme. Takto postupujeme do

té doby, až mřížku otočíme 4x a tím pádem ji dostaneme do původního stavu. Nyní mřížku posuneme o celou svou šířku doprava a vše opakujeme.

Příklad:
Otevřený text: **ŠIFRA!**
Mřížka: $\begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix}$

1. Vezmeme mřížku a zapíšeme první písmeno do otvoru, dostaneme: $\begin{matrix} \text{Š} \end{matrix}$.
2. Mřížku otočíme o 90°ve směru hodinových ručiček $\begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix}$ a zapíšeme opět do prázdného místa druhé písmeno: $\begin{matrix} \text{Š} \\ \text{I} \end{matrix}$.
3. Mřížku otočíme o 90°ve směru hodinových ručiček $\begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix}$ a zapíšeme do prázdného místa třetí písmeno: $\begin{matrix} & \text{Š} \\ \text{F} & \text{I} \end{matrix}$.
4. Mřížku otočíme o 90°ve směru hodinových ručiček $\begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix}$ a zapíšeme do prázdného místa čtvrté písmeno: $\begin{matrix} \text{R} & \text{Š} \\ \text{F} & \text{I} \end{matrix}$.
5. Mřížku otočíme o 90°ve směru hodinových ručiček $\begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix}$ a musíme ji posunout o svou šířku, protože jsme se dostali do výchozí pozice. Zapíšeme do prázdného místa páté písmeno: $\begin{matrix} R & \text{Š} & & A \\ F & I & & \end{matrix}$.
6. Mřížku otočíme o 90°ve směru hodinových ručiček $\begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix}$ a zapíšeme do prázdného místa poslední znak: $\begin{matrix} \text{R} & \text{Š} & & \text{A} \\ \text{F} & \text{I} & & \text{!} \end{matrix}$.
7. Nyní je text transpozičně zašifrován do podoby: $\begin{matrix} \text{R} & \text{Š} & & \text{A} \\ \text{F} & \text{I} & & \text{!} \end{matrix}$

Pro výše uvedený algoritmus napište program, který bude obsahovat:
1. Funkci na načtení mřížky 2x2 a mřížky 4x4 ze souboru. Formát souboru si zvolte sami.
2. Funkci na otočení mřížky o 90°. (Nápověda: transpozice matice + reverse řádků)
3. Funkci na otestování validnosti mřížky. Zdali mřížka po 4x rotaci nezapisuje vícekrát do stejného místa.
4. Funkci na zašifrování textu dle vybrané mřížky.

Pro program si zvolte jazyk PHP, JavaScript nebo Python. Doporučujeme OOP přístup.

4 Dokumentace

Napište do README.md základní dokumentaci, která bude obsahovat tři základní podkapitoly: Statický web, Databáze a Program. V nich stručně seznamte čtenáře s vaším řešením tak, aby čtenář pochopil, co jste měli dělat, jak jste to vyřešili (postupy) a jaký je výsledek práce.
V README.md použijte minimálně: nadpis, podnadpis, odrážkový seznam, obrázek, odkaz, zdrojový kód.
V dokumentaci bude také název práce, autor, rok a měsíc vytvoření.