

|                                |  |
|--------------------------------|--|
| <b>Název úlohy</b>             | <b>Caesarova šifra 1/2</b>   |
| <b>Třída</b>                   | 6. třída   |
| <b>Úloha splňuje rámce</b>     | <ul style="list-style-type: none"> <li>• DATA, INFORMACE A MODELOVÁNÍ – kódování a přenos dat</li> <li>• ALGORITMIZACE A PROGRAMOVÁNÍ – řešení problému krokováním, programování, kontrola řešení</li> </ul>   |
| <b>Propojení s RVP výstupy</b> | <ul style="list-style-type: none"> <li>• <b>I-9-2-05</b> - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za něj; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné</li> </ul> |
| <b>Propojení s ŠVP výstupy</b> | <ul style="list-style-type: none"> <li>• Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program, dbá na jeho čitelnost a přehlednost</li> </ul>  |
| <b>Časová náročnost</b>        | 45 minut (jedna vyučovací hodina)  |
| <b>Stručný popis úlohy</b>     | Žáci se seznámí s šifrováním a pomocí předpřipravených bloků v MakeCode si zkusí zašifrovat jejich vlastní zprávu.   |
| <b>Odkaz na rozšíření</b>      | <a href="https://github.com/microbit-cz/pxt-caesar-cipher-extension">https://github.com/microbit-cz/pxt-caesar-cipher-extension</a>  |

# Caesarova šifra 1/2

## Co je Caesarova šifra?

Caesarova šifra je postavena na principu posunu podle předem zvoleného „klíče“ (tj. číslo). Všechny znaky ve zprávě jsou v tomto procesu zaměněny právě za ty, které se v dané řadě (např. ASCII tabulka) vyskytují na místě hodnoty klíče přičtené k pozici původního znaku.

## Co budete potřebovat

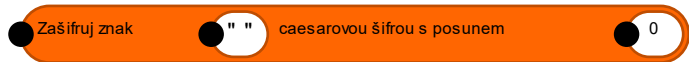

- PC s přístupem k [MakeCode](#)
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit

## Začátek

Tato úloha tvoří první část z dvou. V následující části se žáci seznámí s pojmem šifra a vytvoří algoritmus pro zašifrování a dešifrování Caesarovou šifrou.

## Rozšíření

## Popis rozšíření

|  |  |
|--|--|
| <b>Zašifruj</b><br> | <ul style="list-style-type: none"><li>• Zašifruje zadaný text zadaným klíčem</li><li>• Parametry:<ul style="list-style-type: none"><li>○ znak k zašifrování (znak)</li><li>○ klíč (posun) - může být kladný i záporný (číslo)</li></ul></li><li>• Návrátová hodnota: zašifrovaný znak (znak)</li></ul> |
| <b>Dešifruj</b><br> | <ul style="list-style-type: none"><li>• Dešifruje zadaný text zadaným klíčem</li><li>• Parametry:<ul style="list-style-type: none"><li>○ znak k dešifrování (znak)</li><li>○ klíč (posun) - může být kladný i záporný (číslo)</li></ul></li><li>• Návrátová hodnota: dešifrovaný znak (znak)</li></ul> |

## Vyzkoušení s Microbitem

Tentokrát se jedná o složitější variantu. V rozšíření jsou k dispozici dva bloky, které ale na rozdíl od verze pro 4. ročník šifrují/dešifrují pouze jeden znak, a ne celý řetězec. Úkolem studentů je tedy přijít na to, jak rozdělit jejich zprávu na jednotlivé znaky a poté ji na konci znovu spojit ve výsledný text.

V této úloze budeme pro zjednodušení používat pouze znaky níže uvedené:

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

*Pokud student ve své zprávě zvolí jakýkoliv jiný znak, nezašifruje se (např. mezera zůstává mezerou.)*

*V případě diakritiky samotný microbit sice nevyhodí chybu, ale jelikož ji neumí zobrazovat, ukáže se na displeji místo jiného písmena pouze prázdný znak. Proto znovu žáky upozorněte, aby používali pouze písmena, která mají napsaná na pracovním listě ve **cvičení 3**.*

*Je jedno, jestli žáci používají velká nebo malá písmena. Každé písmeno se vždy automaticky převede na malé a až potom se zašifruje.*

## Možné řešení

1. Vytvoříme si funkci „Zasifruj“, která si jako parametry bere text k zašifrování a posun (klíč).

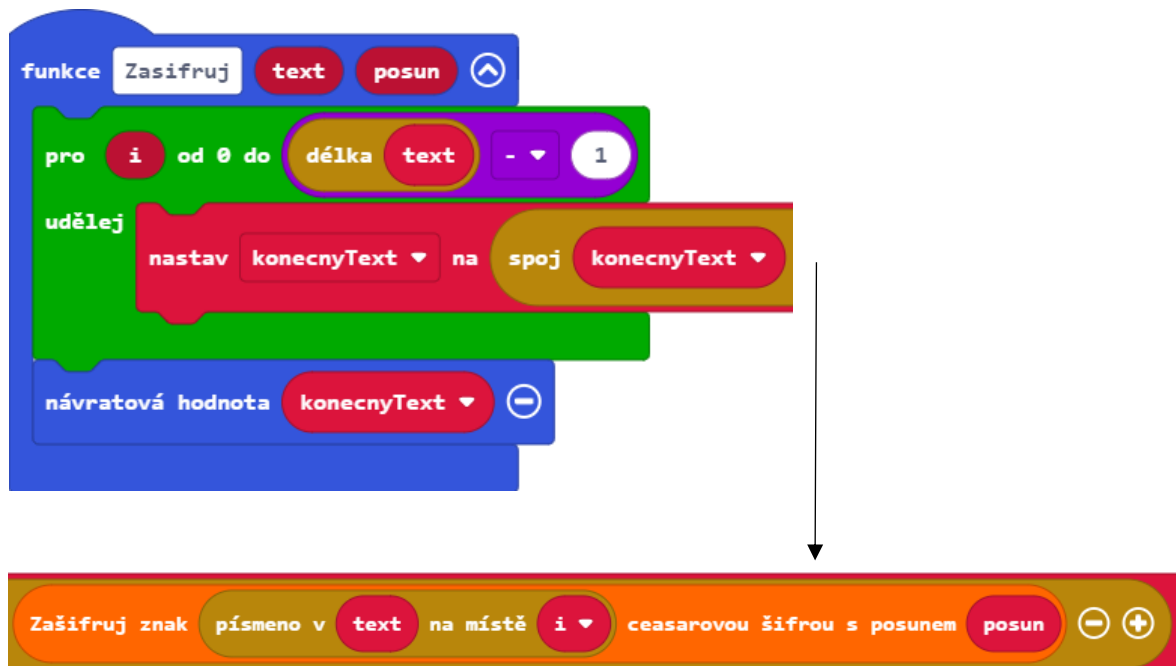
Sekce „funkce“ v prostředí Makecode se skrývá pod možností „Rozšířené“ (je potřeba kliknout na šipku).

Ve funkci nalezneme smyčku s iterační („opakovací“) proměnnou „i“, která bude reprezentovat, na jakém znaku se zrovna nacházíme a z toho důvodu bude nabývat hodnoty 0 (první znak) až délka vloženého textu - 1 (začínáme od 0, proto musíme odečíst 1).

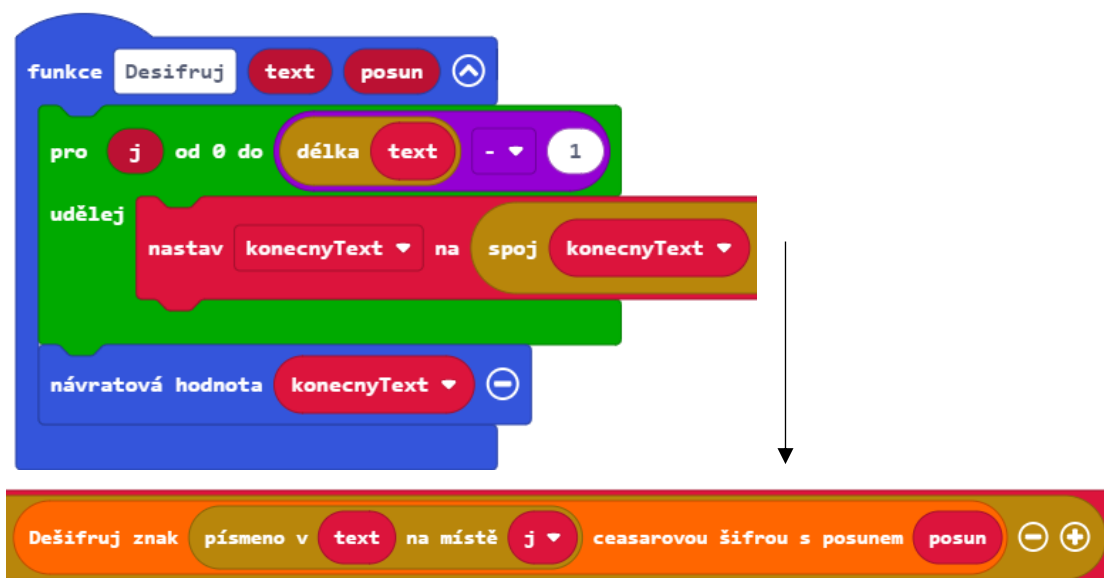
Nyní si vytvoříme proměnnou *konecnyText* (to v blokovém kódu není vidět).

Proměnnou *konecnyText* nastavíme na hodnotu, kterou doteď měla a přidáme k ní zašifrovaný znak, který získáme tak, že si zavoláme z rozšíření blok „Zašifruj znak“. Tomu poté předáme posun (z parametru) a písmeno ve vloženém textu (z parametru) na aktuální pozici (tu máme uloženou v proměnné „i“).

Po provedení celé smyčky jednoduše z naší funkce vrátíme proměnnou *konecnyText*.



2. Dešifrování funguje velice podobně jako zašifrování s tím rozdílem, že nevoláme z rozšíření blok Zašifruj, ale Dešifruj.



3. Nakonec můžeme funkci použít v bloku „Při startu“.

