

# ..... Informator maturalny od 2005 r.

## Arkusz II, zadanie 6.

Szyfr Cezara to szyfr, którego nazwa pochodzi od Gajusza Juliusza Cezara, rzymskiego wodza i polityka. Cezar stworzył go do kodowania swojej korespondencji. Jest to dzisiaj klasyczny przykład szyfru przesuwanego z kluczem  $k = 3$ .

Jego zasada polega na zastąpieniu danej litery literą leżącą o  $k$  pozycji w prawo w stosunku do litery kodowanej. Kodując literę **a**, należy zapisać **d**, zamiast **k** zapiszemy **n**, ale zamiast **y** należy zapisać **b**.

Przyjrzyj się poniższemu tabelom. Zwróć uwagę, że zapisany w tabeli 1 alfabet jawny nie zawiera wszystkich liter alfabetu łacińskiego: brak w nim litery **v**.

tabela 1. Alfabet jawny

| a | b | c | d | e | f | g | h | i | j | k  | l  | m  | n  | o  | p  | q  | r  | s  | t  | u  | w  | x  | y  | z  |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

tabela 2. Alfabet szyfrowy dla  $k = 3$

| d | e | f | g | h | i | j | k | l | m | n  | o  | p  | q  | r  | s  | t  | u  | w  | x  | y  | z  | a  | b  | c  |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Szyfr Cezara można rozszerzyć na dowolny klucz  $k$ , gdzie  $k \in \{1, 2, \dots, 24\}$ .

**Wykonaj poniższe polecenia**, posługując się wcześniej wybranym językiem programowania:

- Napisz funkcję szyfrującą, która dla wprowadzonej litery **l** oraz klucza  $k$  — wyświetli zaszyfrowaną literę.
- Napisz funkcję deszyfrującą, która wyświetli dla wprowadzonej zaszyfrowanej litery **z** oraz klucza  $k$  — literę odszyfrowaną.
- Napisz procedurę (funkcję w C/C++), która wyświetli na ekranie tablicę jawnego alfabetu (patrz tabela 1.) oraz tablicę szyfrowego alfabetu dla dowolnego  $k \in \{1, 2, \dots, 24\}$  (tablice powinny zostać wyświetlone w sposób umożliwiający sprawdzenie działania funkcji szyfrujących).
- Napisz program, który wykorzysta napisane wcześniej funkcje lub procedury:
  - do programu powinny być wprowadzone tylko **małe** litery (program nie musi być odporny na błędnie wprowadzone dane) do zakodowania;
  - program powinien być odporny na błędnie wprowadzoną wartość klucza;
  - układ wyświetlonych wyników powinien być zgodny z układem przedstawionym w poniższej ramce.

*podaj klucz szyfrowania  $k=6$*

*podaj literę do zaszyfrowania  $=y$*

*litera zaszyfrowana to  $e$*

*podaj literę do deszyfrowania  $=g$*

*litera deszyfrowana to  $a$*

*sprawdzenie*

*alfabet jawny*

|          |          |          |          |          |          |          |          |          |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>a</i> | <i>b</i> | <i>c</i> | <i>d</i> | <i>e</i> | <i>f</i> | <i>g</i> | <i>h</i> | <i>i</i> | <i>j</i>  | <i>k</i>  | <i>l</i>  | <i>m</i>  | <i>n</i>  | <i>o</i>  | <i>p</i>  | <i>q</i>  | <i>r</i>  | <i>s</i>  | <i>t</i>  | <i>u</i>  | <i>w</i>  | <i>x</i>  | <i>y</i>  | <i>z</i>  |
| <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> | <i>5</i> | <i>6</i> | <i>7</i> | <i>8</i> | <i>9</i> | <i>10</i> | <i>11</i> | <i>12</i> | <i>13</i> | <i>14</i> | <i>15</i> | <i>16</i> | <i>17</i> | <i>18</i> | <i>19</i> | <i>20</i> | <i>21</i> | <i>22</i> | <i>23</i> | <i>24</i> | <i>25</i> |

*alfabet szyfrowy*

|          |          |          |          |          |          |          |          |          |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>g</i> | <i>h</i> | <i>i</i> | <i>j</i> | <i>k</i> | <i>l</i> | <i>m</i> | <i>n</i> | <i>o</i> | <i>p</i>  | <i>q</i>  | <i>r</i>  | <i>s</i>  | <i>t</i>  | <i>u</i>  | <i>w</i>  | <i>x</i>  | <i>y</i>  | <i>z</i>  | <i>a</i>  | <i>b</i>  | <i>c</i>  | <i>d</i>  | <i>e</i>  | <i>f</i>  |
| <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> | <i>5</i> | <i>6</i> | <i>7</i> | <i>8</i> | <i>9</i> | <i>10</i> | <i>11</i> | <i>12</i> | <i>13</i> | <i>14</i> | <i>15</i> | <i>16</i> | <i>17</i> | <i>18</i> | <i>19</i> | <i>20</i> | <i>21</i> | <i>22</i> | <i>23</i> | <i>24</i> | <i>25</i> |

Pamiętaj, że ocenie podlega też styl programowania (odpowiednie nazywanie zmiennych, stosowanie niezbędnych komentarzy, stosowanie wcięć w zapisie kodu).

**Do oceny oddajesz** plik ..... zawierający kompletny program (z funkcjami i procedurą) napisany w wybranym przez Ciebie języku.