



Round 2 (Opravný A4)

Pravidlá

- Vo výslednom kóde nesmiete používať globálne premenné.
- V súbore `functions.c` sa v hlavnej funkcii `main()` môže nachádzať ľubovoľný kód. Tento nebude síce pri vašich testoch spustený, ale musí byť napísaný správne.
- Pri preklade kódu sú použité nasledovné prepínače: `-std=c11 -Wall -Werror -lm`.
- Pri implementácii môžete použiť ľubovoľnú štandardnú knižnicu, napr. `stdio.h`, `stdlib.h`, `string.h`, `ctype.h`, `stdbool.h` a pod.
- K dispozícii máte aj celý obsah stránky predmetu [Základy algoritmizácie a programovania](#).

Funkcia `find_first_A()` (3b)

Vytvorte funkciu `int find_first_A(const char string[])`, ktorá nájde prvý znak 'A' resp. 'a' v reťazci. Na veľkosti písmen nezáleží.

Input Params

- `const char string[]` - vstupný reťazec

Return

Funkcia vráti hodnotu zodpovedajúcu indexu, na ktorom sa nachádza prvý znak 'A' resp. 'a' v reťazci (na veľkosti nezáleží). Avšak ak vstupný reťazec neobsahuje znak 'A' ani 'a' alebo ak je vstupný reťazec `NULL`, funkcia vráti hodnotu `-1`.

Example (for main() only)

```
printf("%d\n", find_first_A("Tomorrow afternoon"));  
// 9
```

Funkcia `direction_correction()` (3b)

Vytvorte funkciu `int direction_correction(const int degree)`, ktorá upraví stupeň otočenia robota Karla podľa nasledujúcich pravidiel::

1. Každých 90 stupňov predstavuje jeden smer. Základné stupne otočenia: 0-východ, 90-sever, 180-západ, 270-juh.
2. Ak je stupeň otočenia väčší ako základné stupne otočenia, je potrebné ho upraviť, napr. 360 na 0 alebo 450 na 90.
3. Otočenie je neplatné, ak je stupeň otočenia `degree` záporný alebo ak nie je násobkom 90.

Input Params

- `const int degree` - vstupné číslo, počet stupňov

Return

Funkcia vráti hodnotu zodpovedajúcu základnému stupňu otočenia robota Karla. Avšak ak je otočenie neplatné, funkcia vráti hodnotu -1.

Example

```
printf("%d %d %d\n", direction_correction(-90), direction_correction(540),  
direction_correction(180));  
// -1 180 180
```

Funkcia `is_triangle()` (5b)

Vytvorte funkciu `int is_triangle(const int alpha, const int beta, const int gamma)`, ktorá zistí, či je možné zostrojiť trojuholník na základe troch daných uhlov. Platí, že súčet uhlov trojuholníka musí byť 180.

Input Params

- `const int alpha` - vstupná hodnota, uhol trojuholníka
- `const int beta` - vstupná hodnota, uhol trojuholníka
- `const int gamma` - vstupná hodnota, uhol trojuholníka

Return

Funkcia vráti hodnotu `1`, ak je možné zostrojiť trojuholník a hodnotu `0`, ak nie je možné zostrojiť trojuholník. Avšak ak je niektorý z uhlov menší ako `1`, funkcia vráti hodnotu `-1`.

Example (for main() only)

```
printf("%d %d\n", is_triangle(120,30,30), is_triangle(-20,0,200));  
// 1 -1
```