

Základy programování (IZP)

5. počítačové cvičenie

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 602 00 Brno - Královo Pole
ilazur@fit.vut.cz



Zverejnené zadanie 1. projektu

1. Projekt - práca s textom

- **Deadline: 25. 10. 23:59**
- Cieľom je zamyslieť sa nad algoritmizáciou a uložením dát
- Nepotrebuje pracovať s dynamickou pamäťou
- Nepotrebuje pracovať s dočasnými súbormi

Pozor na plagiátorstvo!

Pred odovzdaním si to otestujte na Merlinovi

Príklad vlastnej štruktúry

```
struct date_t {  
    int year;  
    int month;  
    int day;  
};
```

Použitie

```
struct date_t date1;
```

Vnorené štruktúry

```
struct measurement_t {  
    struct date_t date;  
    float temperature;  
};
```

Ako návratový typ

```
struct date_t is_valid_date(struct date_t date);
```

Napište funkciu, ktorá overí platnosť dátumu

```
#include <stdio.h>
struct date_t {
    int year;
    int month;
    int day;
};

int main() {
    struct date_t date;
    printf("Please specify the date (format yyyy mm dd): ");
    scanf("%d %d %d", &date.year, &date.month, &date.day);

    // Check that the date is valid

    return 0;
}
```

Napíšte funkciu, ktorá overí platnosť dátumu

```
int is_valid_date(struct date_t date) {  
    if (date.year < 0 || date.year > 2500 || date.month < 1 ||  
        date.month > 12 || date.day < 1 || date.day > 31) {  
        return 0;  
    }  
  
    if (date.day == 31 && date.month != 1 &&  
        date.month != 3 && date.month != 5 &&  
        date.month != 7 && date.month != 8 &&  
        date.month != 10 && date.month != 12) {  
        return 0;  
    }  
  
    return 1;  
}
```

```
is_valid_date(date);
```

Porovnajte dva dátumy a vypíšte, ktorý je skorší

```
#include <stdio.h>
struct date_t {
    int year;
    int month;
    int day;
};

int main() {
    struct date_t date1, date2;

    printf("Please specify the first date (format yyyy mm dd): ");
    scanf("%d %d %d", &date1.year, &date1.month, &date1.day);
    printf("Please specify the second date (format yyyy mm dd): ");
    scanf("%d %d %d", &date2.year, &date2.month, &date2.day);

    // Compare the dates and find the earlier one

    return 0;
}
```

Porovnajte dva dátumy a vypíšte, ktorý je skorší

```
int earlier_date(struct date_t date1, struct date_t date2) {  
    if (date1.year < date2.year) {  
        return 0;  
    } else if (date1.year > date2.year) {  
        return 1;  
    }  
    if (date1.month < date2.month) {  
        return 0;  
    } else if (date1.month > date2.month) {  
        return 1;  
    }  
    if (date1.day < date2.day) {  
        return 0;  
    } else if (date1.day > date2.day) {  
        return 1;  
    }  
    return 2;  
}
```

```
earlier_date(date1, date2);
```

Zatiaľ pracujeme iba s poliami s pevnou dĺžkou

Inicializácia

```
#include<stdio.h>

#define MX_ROWS 3
#define MX_COLS 3

int main() {
    int arr2d(MX_ROWS)(MX_COLS);

    for (int row = 0; row < MX_ROWS; row++) {
        for (int col = 0; col < MX_COLS; col++) {
            arr2d(row)(col) = 0;
        }
    }

    return 0;
}
```


Nájdite a vypíšte index hľadaného čísla

```
#include<stdio.h>

#define MX_ROWS 3
#define MX_COLS 3

int main() {
    int arr2d(MX_ROWS)(MX_COLS);

    for (int row = 0; row < MX_ROWS; row++) {
        for (int col = 0; col < MX_COLS; col++) {
            arr2d(row)(col) = 0;
        }
    }

    arr2d(1)(2) = 5;
    // Find index of value 5

    return 0;
}
```

Nájdite a vypíšte index hľadaného čísla

```
int find_value(int arr(MX_ROWS)(MX_COLS), int value) {  
    for (int row = 0; row < MX_ROWS; row++) {  
        for (int col = 0; col < MX_COLS; col++) {  
            if (arr(row)(col) == value) {  
                printf("Value %d is on index %d, %d",  
                    value, row, col);  
                return 0;  
            }  
        }  
    }  
    return -1;  
}
```

```
find_value(arr2d, 5);
```

Načítanie

```
// Data type of input stream
FILE *input;

// We attempt to open the file in a read mode
input = fopen("matrix_save.txt", "r");

// We should check that the file operation was
successful
if (input != NULL) {
    char word[101];
    fscanf(input, "%99s", word);
}
// Don't forget to close the file!
fclose(input);
```

Ukládanie

```
// Data type of input stream
FILE *output;

// We attempt to open the file in a write mode
output = fopen("matrix_save.txt", "w");

// We should check that the file operation was
successful
if (input != NULL) {
    fprintf(output, "%d ", 5);
    fprintf(output, "\n");
}
// Don't forget to close the file!
fclose(output);
```

Aký je súčet čísiel vo vstupnom súbore

```
#include<stdio.h>

int main() {
    int count = 0;
    FILE *input;

    input = fopen("text.txt", "r");

    // Make sum of all input numbers
    printf("%d\n", count);

    fclose(input);
    return 0;
}
```

Aký je súčet čísiel vo vstupnom súbore

```
#include<stdio.h>

int main() {
    int count = 0;
    FILE *input;

    input = fopen("text.txt", "r");

    if (input != NULL) {
        int number = 0;
        while (fscanf(input, "%d", &number) != EOF) {
            count += number;
            number = 0;
        }
    }
    printf("%d\n", count);

    fclose(input);
    return 0;
}
```

Stačí vypracovať jednu variantu

- Načítajte zo vstupu 3 merania teploty s dátumom s pomocou predpripravenej štruktúry. Overte správnosť zadaného dátumu. Ak dátum nie je validný, program ukončíte. Vypíšte priemernú teplotu spočítanú zo všetkých meraní. Pri riešení nemusíte použiť pole štruktúr, ale vytvorte aspoň 1 funkciu.
- Vytvorte 2D pole s rozmermi 5 x 5. Inicializujte ho hodnotou 0 a načítajte od používateľa jedno číslo. Na základe tohto čísla vyplňte hodnotou 1 všetky indexy nad diagonálou, alebo pod diagonálou. Následne hodnoty z pola formátovane vypíšte. Vytvorte aspoň jednu funkciu.
- Vytvorte si pomocný textový súbor s ľubovoľným textom o dĺžke maximálne 100 znakov. Načítajte tento súbor do stringu. Spočítajte počet výskytov ľubovoľného písmena. Výsledok vypíšte.