

ЗАДАНИЕ 1

Управлението на функцията `main()` е средство, което позволява на програмите да комуникират с операционната система и да приемат входни данни още при самото им стартиране.

„Конзолен файлов мениджър с аргументи“

Цел на проекта: Да се разработи конзолно приложение в **Dev-C++**, което използва параметрите на функцията `main()`, за да извършва различни операции (събиране на числа, извеждане на текст или проверка на потребителско име).

1. Теоретична основа: Параметрите на `main()`

За да приема данни от командния ред (Command Line Arguments), функцията `main` трябва да бъде дефинирана по следния начин:

C++

```
int main(int argc, char* argv[])
```

- **argc (Argument Count):** Целочислен параметър, който указва броя на подадените аргументи (включително името на самата програма).
 - **argv (Argument Vector):** Масив от низове (`char pointers`), съдържащ самите аргументи. `argv[0]` винаги е името/пътят на програмата.
-

2. Условие на задачата

Напишете програма, която се стартира през конзолата (CMD) и приема точно три аргумента:

1. **Операция:** низ (`add` или `print`).
2. **Стойност 1:** число (ако операцията е `add`) или име (ако е `print`).
3. **Стойност 2:** число (ако операцията е `add`).

Примерен вход в конзолата:

`myprog.exe add 5 10` -> Резултат: Sum is 15

`myprog.exe print Ivan` -> Резултат: Hello, Ivan!

3. Примерен код (Dev-C++)

C++

```
#include <iostream>
```

```
#include <string>
```

```
#include <cstdlib> // За функцията atoi (превръща низ в число)
```

```
using namespace std;
```

```
int main(int argc, char* argv[]) {
```

```
    // 1. Проверка на броя аргументи
```

```
    if (argc < 2) {
```

```
        cout << "Greshka: Lipsvat argumenti!" << endl;
```

```
        cout << "Upotreba: " << argv[0] << " [add/print] [stoinost1] [stoinost2]" << endl;
```

```

    return 1; // Код за грешка
}

string operation = argv[1];

if (operation == "add") {
    if (argc < 4) {
        cout << "Greshka: Za 'add' sa nujni dve chisla!" << endl;
    } else {
        int num1 = atoi(argv[2]);
        int num2 = atoi(argv[3]);
        cout << "Resultat: " << num1 + num2 << endl;
    }
}
else if (operation == "print") {
    if (argc < 3) {
        cout << "Greshka: Vvedete ime!" << endl;
    } else {
        cout << "Zdravei, " << argv[2] << "!" << endl;
    }
}
else {
    cout << "Nepoznata operacia: " << operation << endl;
}

return 0;
}

```

4. Упътвания за работа с Dev-C++

Как да тестваме аргументите в Dev-C++?

Тъй като аргументите се подават при стартиране, има два начина да тествате програмата:

Вариант А (През настройките на средата):

1. Отидете на менюто **Execute -> Parameters...**
2. В полето „Parameters to pass to your program“ напишете аргументите, например: add 20 30
3. Натиснете ОК и след това **Compile & Run (F11)**.

Вариант Б (През системната конзола):

1. Компилирайте програмата (F9).
 2. Отворете папката, където е записан вашият .exe файл.
 3. Задръжте клавиша **Shift** и щракнете с десен бутон в папката -> Изберете **"Open PowerShell/Command window here"**.
 4. Напишете името на файла и аргументите: Project1.exe add 5 7.
-

- **Валидация на argc:** Програмата трябва да проверява дали потребителят е въвел правилния брой данни, преди да се опита да ги достъпи (за избягване на *Segmentation Fault*).
- **Типове данни:** Правилно преобразуване на текстовите аргументи (`argv` са винаги низове) в числа чрез `atoi()` или `atof()`.
- **Логика:** Използване на структури за избор (`if-else` или `switch`) спрямо първия аргумент.