

### Задача 1: Побитови операции

Да се разработи програма тип меню, която обработва присъствия и отсъствия на клас от 64 студенти. За целта да се използва 64-битова маска, а не масив или друга сложна структура от данни!

- да се добави опция за въвеждане на присъствие;
- да се добави опция за въвеждане на отсъствие (изтриване на присъствие);
- да се добави опция за извеждане на номерата на студентите, които отсъстват;
- да се добави опция за извеждане на номерата на студентите, които присъстват;
- да се добави опция за промяна на статута на студент - ако присъства да се запише като отсъстващ или ако отсъства да се запише като присъстващ.

Да се добави валидация на данните - невалидна опция, невалиден номер на студент...

За решението на задачата може (но не е задължително) да се използва следната структура:

```
1. int main(void)
2. {
3.     uint64_t attendance = 0;
4.     int option;
5.     while (1)
6.     {
7.         std::cout << "1. Set attendance" << std::endl;
8.         std::cout << "2. Clear attendance\n" << std::endl;
9.         std::cout << "3. Attendance info\n" << std::endl;
10.        std::cout << "4. Change attendance\n" << std::endl;
11.        std::cout << "5. Exit" << std::endl;
12.        Std::cin >> option;
13.        if (option == 1)
14.        {
15.            // to do
16.        }
17.        else if (option == 2)
18.        {
19.            // to do
20.        }
21.        else if (option == 3)
22.        {
23.            // to do
24.        }
25.        // to do
26.    }
27.    return 0;
28. }
```

### Задача 2:

Да се направи програма, която изчислява лицето и периметъра на кръг с радиус *radius*:

```
int circle(double radius, double* perimeter, double* area)
```

При успех функцията да връща 0, а при неуспех -1 - (напр. отрицателен радиус). Във функцията *main* да се направи така, че данните за кръга да се четат от стандартния вход. Въвеждането да се приключи до въвеждане на **EOF**. При некоректни данни да се изведе съобщение за грешка в стандартния изход за грешки без да се прекратява програмата.

Примерен вход и изход:

```
./circleInfo
```

```
3
```

```
P = 18.85, S = 28.27
```

```
-2
```

```
Invalid input data
```

```
5.1
```

```
P = 32.04, S = 81.71
```

```
EOF (Ctrl+D)
```

### Задача 3:

Да се направи команден калкулатор Като аргументи от командния ред (Command line arguments) се прочита израз от вида:

**[реално число] [операция] [реално число]**

Поддържани операции:

**+** -> събиране,

**-** -> изваждане,

**x** -> умножение,

**/** -> деление.

На стандартния изход да се изведе резултатът от извършената операция. Резултатът да се изведе закръглен до втория знак след десетичната (точка) запетая. В случай на невалиден израз да се изведе съобщение за грешка в стандартния изход за грешки.

Пример:

```
./calculator 10 + 5.2
```

```
Result = 15.20
```

#### Задача 4:

Направете програма **stringCount**, която чете от стандартния вход стрингове до въвеждане на **EOF**. Програмата приема като аргументи от командния ред един аргумент - *C-string* **str**. Програмата да проверява колко пъти е въведен стринга **str** и на стандартния изход да се изведе броя на срещанията.

Примерен вход-изход:

```
./stringCount 'Strypes Bulgaria'
```

```
Hello
```

```
Strypes Bulgaria
```

```
Sofia BG
```

```
Strypes Bulgaria
```

```
1234
```

```
EOF (Ctrl+D)
```

```
Count = 3
```