

Задача 1: Команден калкулатор

Да се направи команден калкулатор. От стандартния вход се прочитат изрази 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.

Пресмятането на изразите за приключи до въвеждане на символа EOF (Ctrl+D).

Пример:

`./calculator2`

`2 + 3.4` -> вход

`5.40` -> изход

`8 - 4.2` -> вход

`3.80` -> изход EOF -> вход (Ctrl+D)

Задача 2: Триъгълник

Да се направи програма, която изчислява лицето и периметъра на триъгълник по дадени три страни.

Да се създаде следната функция:

int triangle(double a, double b, double c, double* S, double *P)

При успех функцията да връща 0, а при неуспех -1.

Програмата да се направи така, че страните за триъгълника да се четат от стандартния вход (на един ред разделени с интервали). Въвеждане да се приключи до въвеждане на **EOF**. При некоректни данни да се изведе съобщение за грешка в стандартния изход.

Примерен вход и изход:

`./triangle`

`10 10 10`

`S = 10.83`

P = 15.00
1 8 -8
Invalid triangle sides!
3 4 5
S = 6.00
P = 12.00

Задача 3: Правоъгълник

Да се направи програма, която изчислява лицето и периметъра на правоъгълни по подадени широчина и височина.

Да се създаде следната функция:

int rectangle(double w, double h, double* S, double *P)

При успех функцията да връща 0, а при неуспех -1.

Програмата да се направи така, че данните за правоъгълника да се четат от стандартния вход (на един ред разделени с интервали). Въвеждане да се приключи до въвеждане на **EOF**. При некоректни данни да се изведе съобщение за грешка в стандартния изход.

Примерен вход и изход:

```
./rectangle
10 10
S = 100.00
P = 40.00
-3 3.1
Invalid rectangle sizes!
3 4
S = 12.00
P = 24.00
```

Задача 4: Квадратно уравнение

Да се направи програма, която решава квадратно уравнение по подадени коефициенти a, b и c.

int quadEq(double a, double a, double c, double* S, double* P)

При успех функцията да връща 0, а при неуспех -1.

Програмата да се направи така, че коефициентите a, b и c да се четат като аргументи от командния ред:

Примерен вход и изход:

```
./quadEquation 1 2 1
```

```
x1 = -1.00
```

```
x2 = -1.00
```

```
./quadEquation 1 4 -5
```

```
x1 = -5.00
```

```
x2 = 1.00
```

```
./quadEquation 10 2 10
```

```
No real roots
```

Задача 5: Побитови маски

unsigned onesCount(unsigned count, size_t size, ...)

Като първи параметър се подава броя на маските, а като втори размера в байтове на маската. (Размерът може да бъде между 1 и sizeof(int)). Функцията да връща общия брой на всички 1-ци в подадените маски.