**Домашна работа - Седмица №3**

**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,

х -> 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:

*./calculator*

*2 + 3.4 -> вход*

*5.40 -> изход*

*8 - 4.2 -> вход*

*3.80 -> изходEOF -> 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(uint64\_t mask)**

Да се направи функция, която преброява колко единици се съдържат в маската **mask.**

**Задача 6: 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. uint32\_t attendance = 0;
4. int option;
5. while (1)
6. {
7. printf("1. Set attendance**\n**");
8. printf("2. Clear attendance**\n**");
9. printf("3. Attendance info**\n**");
10. printf("4. Change attendance**\n**");
11. printf("5. Exit**\n**");
12. scanf("%d", &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. }