

Упражнение 5 - ЕФП
Примерни задачи, за подготовка на КР 1 по ЕФП ЛУ:

УПР 1

1. Да се създаде модул чрез който се изчислява лице на фигура (правоъгълник, квадрат, окръжност, триъгълник, правоъгълен паралелепипед, ромб, конус, триъгълна и четириъгълна пирамида)
2. Да се определи вида на триъгълник по зададени страни/ъгли.
3. Да се създаде приложение „НОК“ на две числа.
4. Сламки трябва да се разпределят в кутии – големи по 50 бр., средни по 20 бр. и малки по 5 бр. Да се създаде функция, която връща броят на кутиите.

УПР 2

1. Да се създаде приложение Лихвен калкулатор, което по зададена сума пресмята проста и сложна лихва за ден, месец и година.

Формули за проста лихва:

$$Kn = Ko * (1 + ((p * n) / 100)) \quad (1.1)$$

Kn – сума за погасяване

Ko - изтеглена сума

p – процент за олихвяване

n – период за година

n = m / 12, където **m** е период в месеци

n = t / B, където **t** е период в дни, **B = 360** дни (приема се, че 1 месец има средно 30 дни)

Формули за сложна лихва:

$$Kn = Ko * (1 + (p / 100))^n \quad (1.2)$$

2. Да се създаде приложение Квадратно уравнение от вида $a * x^2 + b * x + c = 0$. За целта да се използва формула (2.1). Да се пресметнат дискриминантата и квадратните корени на уравнението по зададени стойности за a, b и c. Да се направи проверка за коректни входни данни, да се изведат подходящи съобщения, ако уравнението има два еднакви корена или няма реални корени. Резултата да се закръгли до втори знак след десетичната запетая.

$$x_{1/2} = (-b \pm \sqrt{b^2 - 4 * a * c}) / (2 * a) \quad (2.1)$$

3. Да се създаде приложение Калкулатор изчисляващ 6 аритметични операции между две реални числа: събиране, изваждане, умножение, деление, коренуване, повдигане на степен. При прилагане на определени действия, да се правят проверки и се изведат подходящи съобщения. **Например** проверка за:

- деление на нула (действие деление);
- отрицателно число (действие коренуване).

4. Да се напише програма, която въвежда от клавиатурата цяло трицифрено число. Да се провери дали сумата от цифрите на числото е четно число.
5. Да се напише програма, която въвежда от клавиатурата цяло четирицифрено число. Да се провери дали произведението на цифрите е кратно на 3.
6. Да се напише програма, която въвежда от клавиатурата три цели числа, различни от нула. Програмата да извежда разликата между максималното четно число и минималното нечетно число между тях. Ако въведените числа са само нечетни, да извежда минималното от тях. Ако са само четни – максималното от тях.
7. Да се напише програма, която въвежда от клавиатурата трицифрено число n. Програмата проверява дали цифрите му са различни. Ако са различни, извежда „yes“ и произведението на първата и трета цифра на числото и го отпечатва в обратен ред. В противен случай, извежда „no“.

УПР 3

1. Намерете двуцифрено естествено число ху, за което се въвеждат сборът от цифрите 'x+y' и разликата между числата 'ux - ху'. Да се състави програма на Elixir, която по въведени сбор и разлика извежда цифрите на числото.

Пример: Сбор $x + y = 12$; Разлика $ux - ху = 36$. Търсеното число: 48

2. Да се напише програма, която намира най-голямата и най-малката цифра на дадено неотрицателно цяло число.
3. Да се напише програма, която намира и извежда симетричното на дадено цяло число. Симетрично на дадено число се нарича число със същия знак и същите цифри, по записани в обратен ред.
4. Да се напише програма, която намира сумата на всяко трето цяло число, започвайки от 2 и ненадминавайки 100 (т.е. сумата $2+5+8+11+...+98$).
5. Да се намери сумата от числата в интервала от 0 до 100, които се делят на числото 3.
6. Да се напише функция `increasing_n`, която проверява дали цифрите на дадено естествено число са в нарастващ ред, четени отляво-надясно:

Пример:

за increasing (12489)

Резултат: Да в нарастващ ред са!

за increasing (4456)

Резултат: Не не са в нарастващ ред!

УПР 4

1. Да се напише функция perfect_n, която проверява дали дадено естествено число е съвършено. Съвършени числа са тези, за които сумата на всичките им делители (без самото число) е равна на същото число:

Пример:

за perfect (6)

Резултат: 1+2+3=6

за perfect (28)

Резултат: 1+2+4+7+14=28

за perfect (33550336)

Резултат: 1+2+...+16775168=33550336

2. Символен низ е съставен единствено от малки латински букви. Да се напише програма, която намира и извежда на екрана броя на срещанията на всяка от буквите на низа.

3. Да се напише програма, която определя дали дадено положително цяло число с четен брой цифри на сумата от първите му n цифри е равна на сумата от последните му n цифри ($n = \text{брой_на_цифрите_на_числото} / 2$)

4. Да се напише функция, която намира всички числа в даден интервал от положителни числа в десетична бройна система. Резултатът от тази функция е списък от прости числа, които прочетени и в двете посоки са прости числа. Списъкът да се подреди в нарастващ ред и от него да се изключат числата палиндроми.

Пример: 13 17 31 37 71 73

13 е просто число и прочетено наобратно е 31, което също е просто число.

В интервал от 2 до 100 числата са: 13 17 31 37 71 73 79 97

В интервала от 9900 до 10000 числата са: 9923 9931 9941 9967

5. Да се напише функция increasing_n, която проверява дали цифрите на дадено естествено число са в нарастващ ред, четени отляво-надясно:

Пример:

за increasing (12489)

Резултат: Да в нарастващ ред са!

за increasing (4456)

Резултат: Не не са в нарастващ ред!

