

# Задачи за устния изпит

## Задачи за делимост на числата и други задачи

1. Да се напише програма, която извежда на екрана всички делители на дадено естествено число.

2. Да се напише програма, която извежда на екрана разлагането на прости множители на дадено естествено число.

3. Да се напише функция, която проверява дали дадено естествено число е просто.

4. Като използвате стандартния алгоритъм на Евклид, напишете функция, която пресмята НОД на две естествени числа.

5. Като използвате варианта на алгоритъма на Евклид с изваждане, напишете функция, която пресмята НОД на две естествени числа.

6. Напишете функция, която пресмята НОК за две естествени числа.

7. (Решето на Ератостен)  
[https://en.wikipedia.org/wiki/Sieve\\_of\\_Eratosthenes](https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes)

Напишете програма, която намира всички прости числа от 2 до 3000 с алгоритъма на Ератостен и ги извежда на екрана по 10 на ред.

8. Напишете функция, която реализира бързо повдигане в степен (exponentiation by squaring).

9. Напишете функция, която реализира разширения алгоритъм на Евклид.

10. Напишете програма която отпечатва първите  $n$  реда ( $n \leq 13$ ) от триъгълника на Паскал, като използва връзката:

$$\binom{i}{j} = \binom{i}{j-1} \cdot (i - j + 1) / j, 1 \leq j \leq i, \quad \binom{i}{0} = 1, i = 0, 1, 2, \dots$$

## Задачи за масиви и други задачи

1. Напишете програма за намиране на  $\binom{n}{k}$ , която използва рекурентната връзка  $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ ,  $0 < k < n$ . За целта да се използва масив, в който да се запомни поредния ред от триъгълника на Паскал. Трябва да помним само един ред, защото  $n$ -тия ред се изчислява само като се използва  $n-1$ -вия ред.

2. Да се напише функция, която филтрира масив от цели числа, като премахва елементите му равни на 0. Използва се само паметта на подадения масив.

3. (Последователно търсене)([en.wikipedia.org/wiki/Linear\\_search](http://en.wikipedia.org/wiki/Linear_search))

Да се напише функция, която реализира последователно търсене (Linear search (Sequential search)) на елемент  $x$  в едномерен масив от цели числа. Функцията връща индекса на първия срещнат елемент  $x$  (ако  $x$  е в масива) и -1, ако не е в масива.

4. (Последователно търсене с отрязване)

Да се напише функция, която реализира последователно търсене на елемент  $x$  в сортиран във възходящ ред едномерен масив от цели числа. Функцията връща индекса на първия срещнат елемент  $x$  (ако  $x$  е в масива) и -1, ако не е в масива. Търсенето продължава, докато е сила условието, че не сме стигнали до края на масива и че  $a[i] < x$ .

5. (Binary search)([en.wikipedia.org/wiki/Binary\\_search\\_algorithm](http://en.wikipedia.org/wiki/Binary_search_algorithm))

Да се напише функция, която реализира алгоритъма на двоично търсене на елемент  $x$  в сортиран във възходящ ред едномерен масив от цели числа. Функцията връща някой индекс на  $x$  (ако  $x$  е в масива) и -1, ако не е в масива.

6. (Сортирано сливане). Да се напише програма, която слива сортирано два сортирани във възходящ ред масиви от цели числа в трети масив.

7. Дадени са два сортирани масива от цели числа **a** и **b**. Чрез сливане да се получи трети масив **c**, който съдържа без повторение:

а) всички числа, които принадлежат поне в единия от масивите (обединение на множеството от елементите).

б) всички числа, които принадлежат и на двата масива едновременно (сечение на множеството от елементите).

в) всички числа, които принадлежат на първия масив и не принадлежат на втория масив (разлика на множествата от елементите).

8. Да се напише програма, която реализира метода на разполовяването (Bisection method) за решаването на нелинейното уравнение  $f(x) = 0$  в интервала  $[a, b]$ . Предполагаме, че  $f(x)$  е непрекъсната, строго монотонна и с различни знаци в краищата  $a$  и  $b$  ( $f(a) \cdot f(b) < 0$ ). Методът да се приложи за намирането на реалния корен  $\alpha \in [-2, -1]$  на уравнението  $x^3 - 2x + 2 = 0$ . Коренът да се намери с грешка  $< 10^{-9}$ . За проверка коренът с 9 верни знака е  $\alpha \approx -1.76929235$ .

9. Да се напише функция, която сортира масив от цели числа във възходящ ред с метода на пряката селекция (Selection Sort).

10. Да се напише функция, която сортира масив от цели числа във възходящ ред със сортиране чрез вмъкване (Insertion Sort).

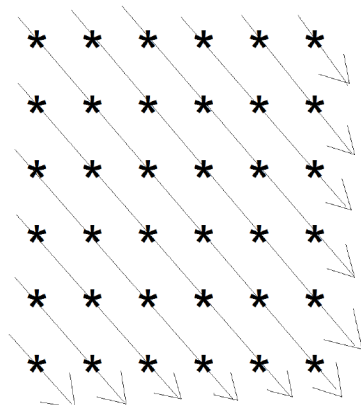
11. Да се напише функция, която сортира масив от цели числа във възходящ ред с метода на мехурчето. (Bubble Sort, Sinking Sort).

12. Задача (Метод на Хорнер)

а) Напишете функция, на която се подава масив от коефициентите на полином  $n$ -та степен  $p(x) = a_0x^n + a_1x^{n-1} + a_{n-1}x + a_n$  и число  $\alpha$  и функцията изчислява по метода на Хорнер коефициентите на полинома от  $n - 1$  степен  $q(x)$ , който е частното на  $p(x)$  при деление на  $x - \alpha$ . Функцията да изчислява и остатъка  $r$  при това деление, който остатък е равен на  $p(\alpha)$ .

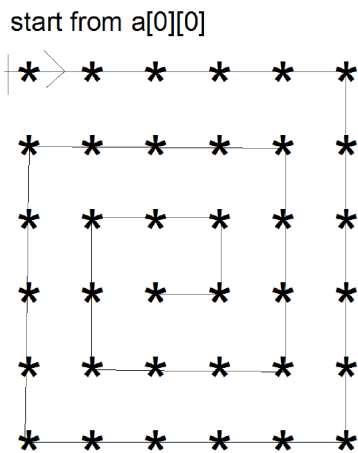
б) Напишете функция, на която се подава масив от коефициентите на полином  $n$ -та степен  $p(x) = a_0x^n + a_1x^{n-1} + a_{n-1}x + a_n$  и число  $\alpha$  и функцията да изчислява по метода на Хорнер  $p(\alpha)$  и  $p(\alpha)'$ .

13. Напишете функция, която отпечатва двумерен масив (матрица) по диагоналите (започвайки отдолу вляво и вървейки нагоре вдясно). Матрицата не е задължително квадратна.



14. Напишете функция, която умножава две матрици. Матриците са правоъгълни със съвместими размери.

15. Напишете функция, която обхожда двумерен масив (матрица) по спирала. Обхождането започва от елемента  $a[0][0]$  в посока по часовниковата стрелка отвън навътре, както е на картинката. Матрицата не е задължително квадратна.



## Задачи за символни низове

Задача 1: Да се напише функция,

а) която проверява дали един символен низ е палиндром. Палиндром е низ, който има симетрия относно средата, първият символ е равен на последния, вторият на предпоследния и така докато символите се засрещнат до средата. Например *level* и *abba* са палиндроми.

б) която проверява дали един символен низ е палиндром, без да се прави разлика между малки и големи букви. Сега и *level*, и *Level* са палиндроми.

Задача 2: Да се напише функция, която изтрива всички срещания на даден символ в даден символен низ .

Задача 3: Да се напише функция, която изтрива повтарящите се интервали в даден символен низ (там където има повече от един интервал, остава по един)

Задача 4:(брой думи в низ) Да се напише функция, която брои думите в даден низ. Дума е последователност от символи, които са английски букви или цифри. Освен това преди тази последователност има символ, който не е буква или цифра или няма символ. След тази последователност има символ, който не е буква или цифра или няма символ. Броят на думите е равен на броят на преминаванията от символ, който не е буква или цифра в символ, който е буква или цифра.

Задача 5:(пример с обхождане дума по дума) Напишете функция, която по даден символен низ връща броя на думите, които съдържат поне веднъж главната буква **A**.

(Забележка: Предложено е решение, което не е директно, а използва един по-общ шаблон ( *Pattern* ), при който се запомня индекса **start** на началото на думата и когато тя завърши при текущия индекс **i**, разполагаме с нейното начало и край)

Задача 6:(масив от низове) Масив от най-много 30 низа, всеки един с дължина най-много 31 се представя с дефиницията **char[30][32]**. Напишете функции, които въвеждат и извеждат такъв масив от низове. Напишете и функция, която сортира лексикографски във възходящ ред елементите на масива, като използва метода на мехурчето.

## Задачи за рекурсия

Задача 1. Напишете рекурсивна функция, която реализира алгоритъма на Евклид за намиране на най-голям общ делител.

Задача 2. Напишете рекурсивна функция, която проверява дали един символен низ е палиндром.

Упътване: Напишете помощна рекурсивна функция, която проверява дали за даден низ поднизът, който започва от позиция  $p1$  до позиция  $p2$  е палиндром. Проверката се декомпозира до проверка на това дали символите на позиция  $p1$  и  $p2$  са равни и дали поднизът от  $(p1 + 1)$ -ва позиция до  $(p2 - 1)$ -ва позиция е палиндром. Простите случаи са когато поднизът е от един символ или е празен, тогава поднизът е палиндром.

Задача 3. Бързо повдигане на степен (бинарен алгоритъм за повдигане на степен). ([https://en.wikipedia.org/wiki/Exponentiation\\_by\\_squaring](https://en.wikipedia.org/wiki/Exponentiation_by_squaring)) Рекурсивната версия на бързото повдигане на степен се основава на формулата:

$$x^n = \begin{cases} 1 & \text{if } n = 0 \\ xx^{n-1} & \text{if } n \text{ is odd, } n > 0 \\ (x^{\frac{n}{2}})^2 & \text{if } n \text{ is even, } n > 0 \end{cases}$$

Бързото повдигане в степен повдига числото  $x$  на цяла неотрицателна степен  $n$  "бързо" - само за брой операции пропорционален на  $\log(n)$ . Направете рекурсивна версия на бързото повдигане на цяло число на цяла неотрицателна степен. В случая рекурсивната версия е по-естествена и за това е по-лесна за реализация от итеративната. Ако си спомняте, за итеративната версия трябваше да построим цикъла с помощта на инварианта.

Задача 4. Бързо повдигане в степен по модул. Основава се на формулата:

$$x^n \% m = \begin{cases} 1 & \text{if } n = 0 \\ ((x \% m)(x^{n-1} \% m)) \% m & \text{if } n \text{ is odd, } n > 0 \\ ((x^{\frac{n}{2}} \% m)(x^{\frac{n}{2}} \% m)) \% m & \text{if } n \text{ is even, } n > 0 \end{cases}$$

Напишете рекурсивна функция, която реализира бързо повдигане в степен по модул. Тествайте програмата за пресмятане на  $595^{703} \pmod{991}$

Отговор 342.

Задача 5. Напишете рекурсивна версия на Binary search algorithm за търсене в сортиран масив. Разгледайте случая на целочислен масив.  
([https://en.wikipedia.org/wiki/Binary\\_search\\_algorithm](https://en.wikipedia.org/wiki/Binary_search_algorithm))

Упътване: Напишете помощна рекурсивна функция, която проверява дали за даден масив подмасивът, който започва от позиция  $p1$  до позиция  $p2$  съдържа елемента  $x$ . Функцията да връща позицията на  $x$  (първата намерена), ако  $x$  е в подмасива и -1, ако  $x$  не е в подмасива. Проверката се свежда до проверка дали средният елемент  $a[mid]$  е равен на  $x$ ,  $mid = (p1 + p2)/2$  и в зависимост от резултата на проверката, евентуално продължава рекурсивно в първата половина - от  $a[p1]$  до  $a[mid - 1]$  или във втората половина - от  $a[mid + 1]$  до  $a[p2]$ . Не забравяйте простия случай, когато подмасивът е празен ( $p1 > p2$ ), тогава функцията връща -1.

Задача 6. Да се напише функция, която рекурсивно намира сумата на елементите на масив.

Задача 7. Да се напише функция, която рекурсивно проверява дали елемент  $x$  принадлежи на масив.

Задача 8. Да се напише функция, която рекурсивно проверява дали елементите на масив са подредени в растящ ред (т.е. дали масивът е сортиран).

Задача 9. Да се напише функция, която рекурсивно проверява дали елементите на масив са различни.