

1. (20 т.) Да се напише функция на C++ `long findSeq(const char* arr, char S)`, която връща като резултат дължината на най-дългата редица от последователни еднакви символи `S` в `arr`.
2. (25 т.) Да се напише функция на C++, `bool findWord(const char* const * grid, size_t M, size_t N, const char* word)`, която проверява дали в матрицата `grid`, с размерност `MxN` се съдържат последователни символи, които образуват думата `word`. Последователни наричаме всички съседни символи по хоризонтала или вертикала в матрицата (без диагонално съседните).
3. (35 т.) Да се напише функция на C++, `int* findSets(const int[] arr, size_t N)`, която приема като параметър масив от положителни числа с дължина `N ∈ (0, 32]`, и връща като резултат нов масив. В него да се съдържат всички непразни множества, чиито елементи са елементи и на `arr`.
Пояснения: Използвайте 0 за разделител между отделните множества, като последното от тях също завършва (терминира се) с две нули. Считаме, че заделената памет ще бъде освободена в извикващата функция.