

Увод в програмирането, 2019-2020 г.

Задачи за домашно № 3

1. Да се направи опростена игра тип “Стани богат”. Програмата да извежда единствен въпрос с 4 възможни отговора, а потребителя избира един от тях. Преди да отговори, потребителят може да поиска жокер (50/50) и в този случай, програмата трябва да елиминира два от грешните отговори. Ако потребителят е дал верен отговор, то той трябва да се информира, че печели, а в противен случай - че губи. След това да се дава възможност за отговор на нов въпрос или изход от играта.
Забележка: Реализирайте опростена “база данни” с въпроси за играта, като глобално дефиниран масив, който съдържа 10 въпроса. Напишете отделна функция, която избира на случаен принцип въпрос от масива (функцията за избор на въпрос **не трябва** да използва `cin` и/или `cout`) и използвайте нея. Няма ограничения за използването на вградени функции за реализация на случайния избор на въпрос.
2. Да се напише програма, която приема вход от потребителя във формат: `<<op X1 X2>>`, където `op` е една от целочислените аритметични операции `[+, -, *]`, а `X1` и `X2` са цели десетични числа с максимален размер 100 цифри. Програмата да извежда на екрана резултата от аритметичната операция.
Забележка: Числата може да са както положителни, така и отрицателни.
3. Езикът на планетата Абецедея се състои от думи, които се образуват според следните правила:
 - Празният низ е дума от езика;
 - Ако `<w>` е дума от езика, то `AA<w>B` също е дума от езика;
 - Ако `<w>` е дума от езика, то `<w>C<w>` също е дума от езика;
 - Ако `<w1>` и `<w2>` са две различни думи от езика, то `<w1>C<w2>` също е дума от езика;
 - Ако `<w>` е дума от езика, то `<w>D<w>` също е дума от езика;
 - Ако `<w>` се състои от повече от 64 символа, то `<w>` не е дума от езика.

Абецедейците са много любопитни и всеки път когато видят някаква последователност от символи се чудят дали това е дума от техния език. Помогнете им като напишете функция с име `isWord()`, която получава като параметър низ и проверява дали той е дума от гореописания език. Напишете функция `main()`, в която от стандартния вход се въвежда цяло число `n`, следвано от `n` реда, на всеки от които има по един низ (празният низ се задава с празен ред). Програмата трябва да извежда на стандартния изход `n` реда, всеки от които съдържа по една цифра: 1, ако подадената на входа дума е дума от езика, или 0 - в противен случай.

Ограничения:

- $1 \leq n \leq 10$
- Дължината на никой от въведените низове не надвишава 255.

Примери:

Вход	Изход	Забележка
5 AABCAAB DD AABD CAAB DDD	1 0 0 1 1	n = 5
10 AAAABB AACB CCCC CCDCCCAAB AABAAB CAABDAAB AABCDAAAB CXC AABAABBB	1 1 1 1 0 1 0 1 0 0	n = 10

Пояснения:

1. Задачи 1 и 2 носят по 3 точки, а задача 3 – 4 точки.
2. В решенията на дадените задачи **не се допуска** използването на String и/или STL библиотеки, както и вградени функции за преобразуване на символ в число и обратно.
3. Всички задачи ще бъдат проверени автоматично за преписване. Домашните с голямо съвпадение ще бъдат проверени ръчно от преподавателите и при установено плагиатство ще бъдат анулирани.
4. Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC.
5. Всяка задача от домашното трябва да бъде решена в точно един, отделен файл. Името на файла трябва да бъде в следния формат:

fnXXXXX_d3_N_CC.cpp, където:

- XXXXX е вашият факултетен номер
 - N е номерът на задачата
 - CC указва кой компилатор сте използвали. Стойността му може да бъде "gcc" за GCC или "vc" за Visual C++.
6. Архивирайте всички файлове, които предавате в един архивен файл, компресиран в стандартен zip формат, със следното име:

UP_19-20_fnXXXXX_d3.zip, където XXXXX е вашият факултетен номер

7. Файловете с решенията, които предавате трябва да са оформени съгласно добрите практики за оформяне на кода, за които се говори по време на лекции и упражнения. Ще се отнемат точки за неинформативни имена на променливи, неизползване на подходящи константи и т.н.
8. Всички предадени програми трябва да се държат адекватно при некоректни входни данни от потребителя. (например въвеждане на поредица от символи, когато програмата очаква число).
9. Файловете с решенията може да съдържат само стандартните символи с кодове от 0-127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
10. Първото нещо във всеки от файловете, които предавате, трябва да бъде коментарен блок, който носи информация за съдържанието на файла. Този коментар трябва да изглежда точно така, както е показано по-долу, като в него попълните своите лични данни. За улеснение, просто копирайте дадения по-долу блок и попълнете в него нужната информация. Обърнете внимание, че на първия ред след наклонената черта има две звезди и че във файловете не може да се съдържат символи на кирилица.

```
/**
 *
 * Solution to homework assignment 3
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia
 * University
```

```

* Winter semester 2019/2020
*
* @author <вашето име>
* @idnumber <вашият факултетен номер>
* @task <номер на задача>
* @compiler <използван компилатор - GCC или VC>
*
*/

```

Например един попълнен блок за студент с име Иван Иванов, ф.н. 12345, който предава задача 2, компилирана с GCC, трябва да изглежда така:

```

/**
*
* Solution to homework assignment 3
* Introduction to programming course
* Faculty of Mathematics and Informatics of Sofia
University
* Winter semester 2019/2020
*
* @author Ivan Ivanov
* @idnumber 12345
* @task 2
* @compiler GCC
*
*/

```

11. Предадени домашни, които не отговарят на условията от точки 2-10 ще бъдат оценени с 0 точки.