**СУ „Св. Климент Охридски“, ФМИ**

Специалност „Софтуерно Инженерство“

**Увод в програмирането, 2021-2022 г.**

**Задачи за домашно № 2**

1. Напишете функция ***bool*** *hasAlternatingBits****(****unsigned n****)***, която приема като параметър естествено число n и връща като резултат *true*, ако в двоичния запис на n няма последователни еднакви битове и *false* - в противен случай. Реализирайте програма, която изпълнява функцията *hasAlternatingBits* върху прочетено от стандартния вход естествено число и извежда на екрана резултата от изпълнението ѝ. Програмата следва да проверява въведеното от клавиатурата число за коректност, преди да го подаде на функцията.

**Пример:**

| **Вход** | **Изход** | **Обяснение** |
| --- | --- | --- |
| 5 | *true* | Двоичното представяне на 5 е: **101** |
| 7 | *false* | Двоичното представяне на 7 е: **111** |
| 1 | *true* | Двоичното представяне на 1 е: **1** |
| 0 | *true* | Двоичното представяне на 0 е: **0** |

1. Да се напише програма, която прочита от клавиатурата естествено число N [1, 109] и извежда на конзолата най-близкото до N число, което е палиндром. Ако има два равно отдалечени палиндрома, програмата да връща по-малкия.

**Уточнение:** Палиндроми наричаме числа с поне 3 цифри, които са еднакви прочетени отляво надясно и отдясно наляво. Намереният от програмата палиндром може да е както, по-голям, така и по-малък от N.

**Пример:**

| **Вход** | **Изход** |
| --- | --- |
| 121 | 121 |
| 56 | 101 |
| 18595 | 18581 |

1. Разполагате с повреден калкулатор, който има само две работещи функции: (1) да умножава по две и (2) да изважда едно. По въведени от клавиатурата начално число и окончателен резултат, изведете в конзолата най-малкият брой операции с така описания калкулатор, за които може да се достигне от началното число до окончателния резултат.

**Пример:**

| **Вход** | **Изход** | **Обяснение** |
| --- | --- | --- |
| 2 3 | 2 | Първо умножаваме по две и после намаляваме с едно (**2 -> 4 -> 3**) |
| 5 8 | 2 | Първо умножаваме по две и после намаляваме с едно (**2 -> 4 -> 3**) |
| 3 10 | 3 | Умножаваме по две, намаляваме с едно и после пак умножаваме по две (**3 -> 6 -> 5 -> 10**). |

1. Да се напише програма, която приема като вход от клавиатурата две цели числа - n и k и отпечатва в конзолата минималния брой от описаните по-долу операции, които трябва да се приложат за да се получи k от n.  
   Валидните операции са:

* Добавяне на цифра в края на числото.
* Премахване на цифра от края на числото.
* Размяна на местата на две цифри в числото.

**Пример:**

| **Вход** | **Изход** | **Обяснение** |
| --- | --- | --- |
| 1234 43215 | 3 | 1234 -> **4**23**1 ->** 4**32**1 -> 4321**5** |
| 987123  123 | 6 | 987123 -> 98**3**12**7** -> 9**2**3**1**87 -> **1**23**9**87 -> 12398 -> 1239 -> 123 |

1. *UNIX време* се нарича броя изминали секунди, считано от 01.01.1970. По подадено като вход число от тип *unsigned int,* което се интерпретира като брой секунди*,* да се изведат в конзолата следните данни:

**ден.месец.година час:минути:секунди**

спрямо [GMT](https://en.wikipedia.org/wiki/Greenwich_Mean_Time). За задачата не е позволено да използвате библиотеки за обработка на timestamp.

**Пояснение:** *Може да използвате следния ресурс, за да тествате решението си: https://www.epochconverter.com/*

**Пример:**

| **Вход** | **Изход** |
| --- | --- |
| 34 | 01.01.1970 00:00:34 |
| 1234567890 | 13.02.2009 23:31:30 |

**Пояснения:**

1. Всички задачи носят по **2** точки.
2. Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно от лектора и при установено плагиатство ще бъдат анулирани.
3. Опитайте се да напишете максимално ефективен код, както по отношение на брой редове, така и по отношение на време за изпълнение. Помислете къде може да се намали броят на повторенията на циклите или да се намали броят на променливите, които използвате за решаване на задачата.
4. За решаване на задачите не се допуска използването на символни низове.
5. Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC
6. Всяка задача от домашното трябва да бъде решена в точно един, отделен файл. Името на файла трябва да бъде в следния формат:

fnXXXXXXXXXX\_d2\_N\_CC.cpp, където:

* XXXXXXXXXX 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++.

1. Архивирайте всички файлове, които предавате в един архивен файл, компресиран в стандартен zip формат, със следното име:

UP\_20-21\_fnXXXXXXXXXX\_d2.zip, където XXXXXXXXXX е вашият факултетен номер

1. 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.
2. Всички предадени програми трябва да следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата. При некоректни входни данни програмата да извежда на екрана -1.
3. 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!).
4. 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 информация за Вас. За улеснение, просто копирайте дадения по-долу блок и попълнете в него необходимите данни, вместо текста, маркиран с ъглови скоби. 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 2

\* Introduction to programming course

\* Faculty of Mathematics and Informatics of Sofia University

\* Winter semester 2021/2022

\*

\* @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 2

\* Introduction to programming course

\* Faculty of Mathematics and Informatics of Sofia University

\* Winter semester 2021/2022

\*

\* @author Ivan Ivanov

\* @idnumber 12345

\* @task 2

\* @compiler GCC

\*

\*/

1. Предадени домашни, които не отговарят на условията от точки 4-11 ще бъдат оценени с 0 точки.