



IX ОЛИМПИАДА ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ И КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ 11 КЛАСС

Задача 1. Генерация пароля.

Агент секретной службы Смит – начальник отдела по сбору секретных и компрометирующих материалов по западному региону, в котором для хранения информации создан специальный закрытый ресурс *PiggyLeaks.ru*, доступ к которому осуществляется с использованием пароля. Каждому сотруднику отдела выдается свой уникальный пароль доступа к материалам. При генерации паролей агент Смит ввел следующие ограничения:

- пароль состоит из 10 цифр, каждая из которых может принимать значение от 1 до 6 включительно;
- сумма любых трех соседних цифр в пароле равна 10.

Помогите агенту Смигу написать программу по генерации паролей для своих сотрудников. Сколько всего сотрудников у него может работать в отделе?

Задача 2. Алгоритм.

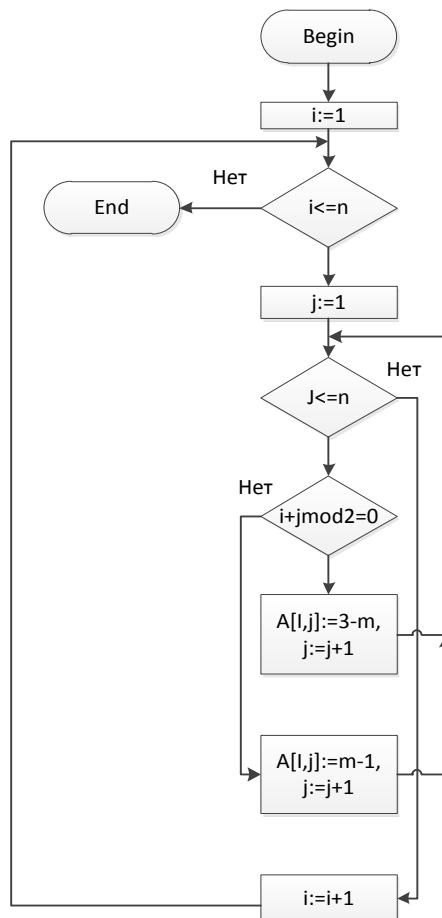


Рис. 1. Блок схема алгоритма обработки матрицы

Квадратную матрицу размером n на n заполнили целыми числами по алгоритму, представленному на блок-схеме (см. рис. 1). При обращении к элементам массива переменная i обозначает номер строки, а переменная j – номер столбца. Индексация начинается с единицы. Найдите минимальное целое положительное значение m , при котором сумма элементов в матрице будет равняться 161, если $n = 13$?

Задача 3. Стеганография.

Реализовать приложение, извлекающее секретное сообщение из входного файла. Известно, что для скрытного внедрения данных используется регистр символа. Если буква в нижнем регистре – это соответствует «0», если буква в заглавном регистре – это соответствует «1». Очередной символ секретного сообщения составляется из 8-ми бит, которые формируют код этого символа. В шифровании используются только буквы русского или английского алфавита. Знаки препинания и цифры не учитываются.

Указание:

Приложение разрабатывается на базе реализованного шаблона чтения из файла. Для получения шифрованного текста необходимо вызвать функцию:

```
void GetCryptoText(char *massiv, int *resultlen;
```

`massiv` – указатель на массив символов, который будет заполнен сообщением после возврата из функции (не менее 500 байт);

`resultlen` – указатель на целочисленную переменную, которая будет равна количеству записанных в `massiv` во время выполнения функции байт.

Декодированное сообщение необходимо вывести на консоль.

Для проверки символа, является ли он буквой, используйте функцию

```
int iswalph(unsigned char c);
```

Для перевода символов в верхний и нижний регистр используйте функции

```
int towupper(unsigned char c);
```

```
int towlower(unsigned char c);
```

Для корректной работы строки, содержащие русские буквы, должны быть объявлены как `unsigned char`.

Заметим, что если вы работаете со средой обработки *VisualStudio*, то необходимо обратить внимание на настройку проекта (правой кнопкой на проекте, «Свойства»): параметр «*CharacterSet*» должен быть установлен в «*UseMulti-ByteCharacterSet*».

Задача 4. Провал.

В штате секретной службы состоят 10 агентов (под номерами 1, 2, ..., 10). Для связи с ними при проведении разведывательной операции используются устройства, которые работают в заданном диапазоне частот, но в них можно настроить индивидуально интенсивность передачи сигнала в минуту (число сигналов в минуту). В случае провала агент отключает передатчик. В штате стоит приёмное устройство, которое считает общее количество пришедших в минуту сигналов от всех агентов. Как надо задать частоты передатчиков, чтобы в штате в случае провалов агентов можно было бы определить их номера.

Задача 5. Шифрование.

Иван написал Егору сообщение и закодировал, используя таблицу 1.

Таблица 1

| № | символ | код | № | символ | код | № | символ | код |
|----|----------|---------|----|----------|---------|----|----------|--------|
| 1 | <i>а</i> | 010 | 12 | <i>к</i> | 100 | 23 | <i>х</i> | 110101 |
| 2 | <i>б</i> | 000 | 13 | <i>л</i> | 0111100 | 24 | <i>ц</i> | 110110 |
| 3 | <i>в</i> | 011000 | 14 | <i>м</i> | 0111101 | 25 | <i>ч</i> | 110111 |
| 4 | <i>г</i> | 011001 | 15 | <i>н</i> | 0111110 | 26 | <i>ш</i> | 111000 |
| 5 | <i>д</i> | 101 | 16 | <i>о</i> | 0111111 | 27 | <i>щ</i> | 111001 |
| 6 | <i>е</i> | 011010 | 17 | <i>п</i> | 110000 | 28 | <i>ъ</i> | 111010 |
| 7 | <i>ё</i> | 011011 | 18 | <i>р</i> | 001 | 29 | <i>ы</i> | 111011 |
| 8 | <i>ж</i> | 0111000 | 19 | <i>с</i> | 110001 | 30 | <i>ь</i> | 111100 |
| 9 | <i>з</i> | 0111001 | 20 | <i>т</i> | 110010 | 31 | <i>э</i> | 111101 |
| 10 | <i>и</i> | 0111010 | 21 | <i>у</i> | 110011 | 32 | <i>ю</i> | 111110 |
| 11 | <i>й</i> | 0111011 | 22 | <i>ф</i> | 110100 | 33 | <i>я</i> | 111111 |

В результате была получена последовательность бит открытого текста $O(i)$, $i=1, \dots, 33$. Затем произвёл преобразование $S(i) = (O(i) + S(i - 1)) \bmod 2$, $S(0)=0$ (см. рис. 2). В результате чего была получена последовательность бит шифрованного текста $S(i)$, $i=0, \dots, 33$:

0011111110011000011001100000001100.

Какое сообщение Иван написал Егору?

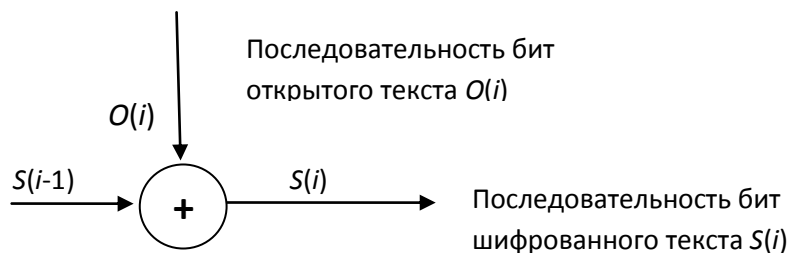


Рис. 2. Схема преобразования закодированного сообщения