

Условия задач заключительного этапа

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.

Квадратную матрицу размером n на n заполнили целыми числами по алгоритму, представленному на блок-схеме (см. рис. 2). 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 $n = 13$?

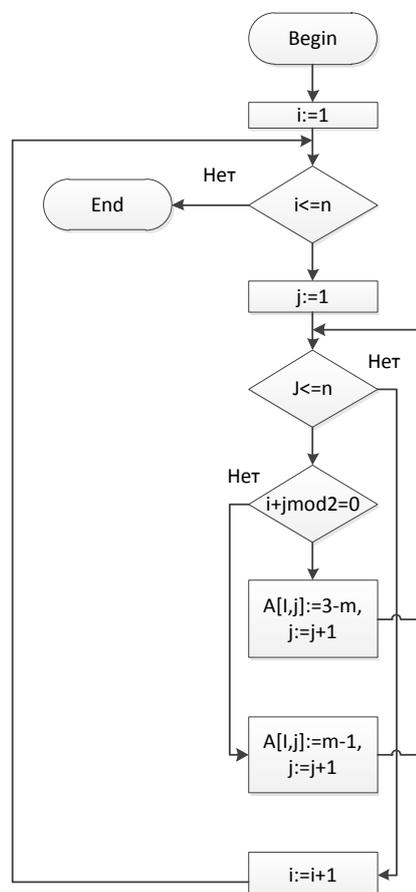


Рис. 2. Блок-схема алгоритма

Задача 3. Стеганография.

Известно, что в текстовом файле «stego.txt» скрыто сообщение. Для его внедрения использовался регистр символа: буква в нижнем регистре соответствует «0», в заглавном регистре – «1». Сообщение представляется восьмиразрядными кодами символов (ASCII), составленными из последовательно считанных бит. Для внедрения используются только буквы русского и английского алфавита. Реализуйте приложение, извлекающее скрытое сообщение из файла.

Указание:

Приложение разрабатывается на базе реализованного шаблона, находящего в папке с заданием.

Для получения текста из файла необходимо вызвать функцию:

```
void GetStegoText(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.

Для проверки является ли символ буквой, используйте функцию:

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

Для перевода символов в верхний и нижний регистр используйте функции:

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

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

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

Заметим, что если вы работаете со средой обработки *Visual Studio*, то необходимо обратить внимание на настройку проекта (правой кнопкой на проекте, «Свойства»): параметр «*Character Set*» должен быть установлен в «*Use Multi-Byte Character Set*».

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 штабе стоит приёмное устройство, которое считает общее количество пришедших в минуту сигналов от всех агентов. Как надо задать частоты передатчиков,

чтобы в штабе 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. 3). В результате чего была получена последовательность бит шифрованного текста $S(i)$, $i=0, \dots, 33$:

0011111110011000011001100000001100.

Какое сообщение Иван написал Егору?

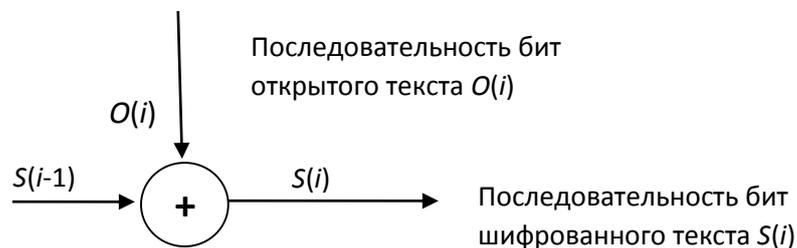


Рис. 3. Схема преобразования закодированного сообщения