



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

Решение.

Представим последовательность цифр в виде: $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$. Покажем, что в последовательности будут повторяться первые три цифры.

Рассмотрим первую тройку цифр: $x_1x_2x_3$. Зафиксируем x_1 и x_2 . Тогда по условию $x_3 = 10 - x_1 - x_2$.

Рассмотрим вторую тройку цифр: $x_2x_3x_4$. x_2 и x_3 подставим из предыдущего шага. Найдем x_4 : $x_4 = 10 - x_2 - x_3 = 10 - x_2 - (10 - x_1 - x_2) = x_1$.

Аналогично x_5 : $x_5 = 10 - x_3 - x_4 = 10 - (10 - x_1 - x_2) - x_1 = x_2$.

В итоге получим последовательность вида $x_1x_2x_3x_1x_2x_3x_1x_2x_3x_1$.

Задача сводится к перебору x_1 и x_2 таких, что $10 - x_1 - x_2 \geq 1$ и $10 - x_1 - x_2 \leq 6$.

Это реализуется двумя вложенными циклами от 1 до 6 включительно.

```
#include<stdio.h>
#include<iostream>
using namespace std;
int main()
{
    int i, j, k;
    int total = 0;
    for (i = 1; i <= 6; i++)
    {
        for (j = 1; j <= 6; j++)
        {
            k = 10 - i - j;
            if (k <= 6 && k >= 1)
            {
                cout<<i<<j<<k<<i<<j<<k<<i<<j<<k<<i<<endl;
                total++;
            }
        }
    }
}
```

```

    }
}
cout << endl << " Всего:: " << total << "комбинаций" << endl;
return 0;
}

```

В результате работы программы переменная *total* будет содержать количество сотрудников, которые могут работать у Смита и иметь различные пароли.

Ответ: 27 сотрудников.

Задача 2. Алгоритм.

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?

Решение.

Согласно представленному алгоритму, матрица заполняется слева направо сверху вниз, начиная со строки с номером 1 и столбца с номером 1 числами вида $x = (3 - m)$ и $y = (m - 1)$ в зависимости от чётности суммы номеров строки и столбца очередного элемента матрицы. После заполнения матрица имеет следующий вид:

- 1-ая строка: хухухухухухух
- 2-ая строка: ухухухухухуху
- ...
- 13-ая строка: хухухухухухух

Обозначим сумму элементов матрицы через *S*. Тогда, с одной стороны, по условию $S=161$, с другой стороны, исходя из описанного выше вида матрицы,

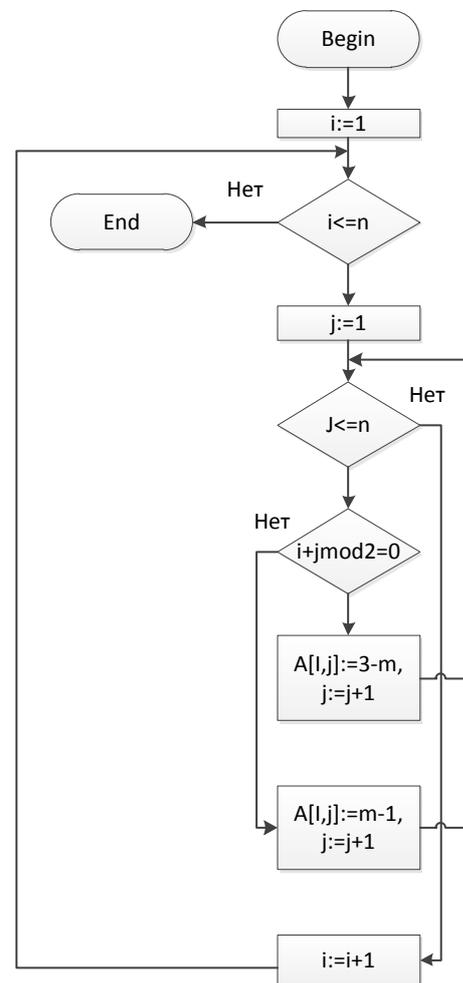


Рис. 1. Блок схема алгоритма обработки матрицы

$$S = 7(7x + 6y) + 6(6x + 7y) = 49x + 42y + 36x + 42y = 85x + 84y = 85(3 - m) + 84(m - 1) = 255 - 85m + 84m - 84 = 171 - m \Rightarrow 161 = 171 - m, \text{ откуда получаем, что } m = 10.$$

Ответ: $m=10$.

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 toupper(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 *VisualStudio*, то необходимо обратить внимание на настройку проекта (правой кнопкой на проекте, «Свойства»): параметр «*CharacterSet*» должен быть установлен в «*UseMulti-ByteCharacterSet*».

Решение.

Переберем все элементы массива кодированных символов и запишем в новый массив «1», если символ в верхнем регистре, иначе запишем «0».

```
char *openmass=new char[size+1];
for(int i=0;i<size;i++)
{
    if ( toupper(cryptomass[i]) == cryptomass[i])
        openmass[i]=1;
    else
        openmass[i]=0;
}
```

Наложим ограничивающее условие «Знаки препинания и цифры не учитываются», что производится установкой блокирующего условия или цикла, пропускающего очередной символ кодированного массива в случае, если он является знаком препинания или цифрой:

```
while(!iswalph((unsigned char) cryptomass[i]) && i<size)
    i++;
```

Таким образом, получаем рабочий цикл, переводящий массив кодированных байт в массив из 0 и 1, представляющих из себя символы русского алфавита, записанные в двоичном представлении. Последним действием по конструированию алгоритма и написанию программы является запись открытого текста в виде привычных символов, что производится с помощью сдвиговых операций. Необходимо пройти выходной массив один раз от начала до конца, записывая 0 и 1 по 8 элементов в один байт с использованием, к примеру, вот такого выражения:

```
openmass[l] |= 1 << (7-j);
```

Здесь j – это инкрементная переменная, принимающая значения от 0 до 7 и позволяющая с помощью операции сдвига сместить двоичную единицу последовательно по всем возможным значениям в байте (1, 2, 4, 8, 16, 32, 64, 128). Слева от равенства используется знак поразрядного сложения, задача которого установить в результирующем байте декодированной последовательности битовую единицу в соответствующее место (см. рис. 2).

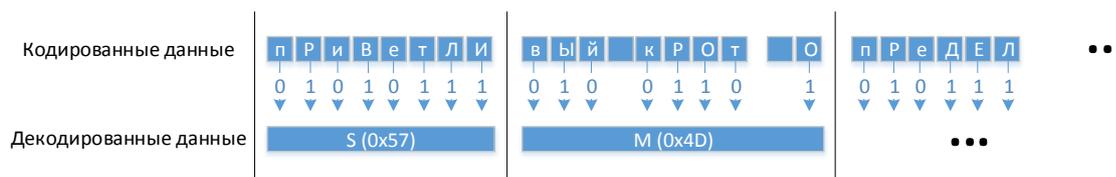


Рис. 2. Сдвиговые операции

Задача решена, но для максимальной оценки необходимо провести оптимизацию, чтобы устранить лишнее преобразование из массива байт в массив 0 и 1, а затем обратно в массив байт. Получается, что входной массив преобразуется в меньший в 8 раз по объёму (если не учитывать знаки пунктуации и цифры). Значит необходимо создать выходной массив сразу и заполнять его последовательно в байтовом представлении. Рабочий цикл приведен ниже:

```
int _tmain(int argc, _TCHAR* argv[]) {
    setlocale(LC_ALL, "Russian");
    unsigned char cryptomass[5000];
    int cryptomasssize=0;
    //////////////////////////////////////
    GetCryptoText(cryptomass, &cryptomasssize);
    //Запишите своё решение ниже
    //...
    char *openmass;
    int size=strlen((char*)cryptomass);
    unsigned char mask=1;
    int k=0, l=0, t=0;
    openmass=new char[size/8+1];
    for(int i=0; i<size; i++) {
        openmass[l]=0;
        for(int j=0; j<8; j++) {
            while(!iswalph((unsigned char)cryptomass[i]) && i<size)
                i++;
            if ( toupper(cryptomass[i]) == cryptomass[i])
                openmass[l] |= 1 << (7-j);
            i++;
        }
        l++;
    }
}
```

```

i--;
}
openmass[size/8]='\0';
printf("%s\n", openmass);
////////////////////////////////////
return 0;
}

```

Ответ: Поздравляю Вас с успешным выполнением задания, Юстас!

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.

Решение.

В штате каждую минуту получают информацию о суммарном числе сигналов N . Представим N в виде сумм степеней двойки

$$N = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_12 + a_0.$$

В силу свойств позиционных систем счисления коэффициенты $a_{n-1}, a_{n-2}, \dots, a_1, a_0$ определяются однозначно для N . Если агент с номером i настроит свой передатчик на передачу 2^{i-1} в минуту, то по коэффициентам представления суммарного числа сигналов N в двоичной системе счисления легко можно определить действующих и провалившихся агентов. Если коэффициент a_j равен 0, то агент провалился, если 1 – действует.

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?

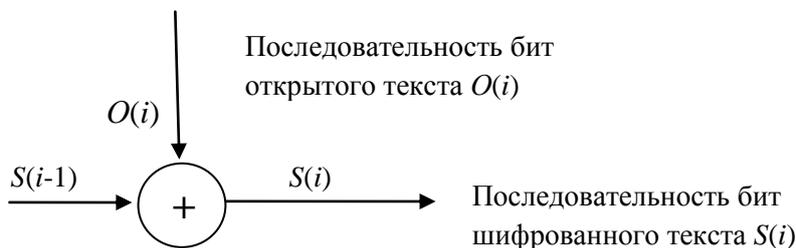


Рис. 3. Схема преобразования закодированного сообщения

Решение.

Согласно схеме шифрования i -ый бит исходного открытого сообщения $O(i)$ равен сумме $(S(i) + S(i-1)) \bmod 2$ для $i > 0$. Учитывая, что $S(0) = 0$ и $S(1) = O(1)$ получаем последовательность бит открытого сообщения $i=1, \dots, 33$

010000001010100010101010000001010.

Воспользуемся таблицей кодов. Ни один символ не закодирован 0 и 01. Последовательность бит 010 соответствует символу *а*. Последующие три бита 000 – символу *б*. Рассуждая аналогично, получаем:

- 010 – *а*
- 000 – *б*
- 001 – *р*
- 010 – *а*
- 100 – *к*
- 010 – *а*
- 101 – *д*
- 010 – *а*
- 000 – *б*
- 001 – *р*
- 010 – *а*,

Таким образом, было закодировано слово *абракадабра*.

Ответ: абракадабра