

XII Межрегиональная олимпиада школьников по информатике и компьютерной безопасности (2016-2017 учебный год)

9-10 КЛАССЫ

Условия задач отборочного этапа

Задача 1

Было перехвачено два сообщения, зашифрованных одним и тем же способом:

1) 11 0F 13 10 29 1F 19 06 17 0E 01 20 02 06 08 0F 16 01 18 0E 15 12 19 1D

2) 14 14 13 06 17 01 23 09 32

Известно, что при шифровании использовалось несколько систем счисления. Первое сообщение удалось расшифровать – «КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ».

Расшифруйте второе сообщение.

Комментарий: При написании ответа использовать только ПРОПИСНЫЕ буквы! При шифровании предполагалось, что Ё=Е.

Задача 2

В классе 100 компьютеров объединены в компьютерную сеть, которая имеет вид прямоугольной сетки размера 10 x 10. Вирус с зараженного компьютера может распространяться на соседние, если заражены два или более непосредственно с ним связанных компьютера. Какое минимальное количество компьютеров надо заразить изначально, чтобы вся сеть оказалась зараженной?

Задача 3

Даны различные функции сортировки массива целых чисел.

<pre>void sort1(int* mass, int size) { int temp; for (int i = 0; i < size - 1; i++) { for (int j = 0; j < size - i - 1; j++) { if (mass[j] > mass[j + 1]) { temp = mass[j]; mass[j] = mass[j + 1]; mass[j + 1] = temp; } } } }</pre>	<pre>void sort2(int* mass, int size) { int key = 0; int temp; for (int i = 0; i < size - 1; i++) { key = i + 1; temp = mass[key]; for (int j = i + 1; j > 0; j--) { if (temp < mass[j - 1]) { mass[j] = mass[j - 1]; key = j - 1; } } } }</pre>
---	--

<pre> } } </pre>	<pre> } mass[key] = temp; } } </pre>
<pre> void sort3(int *mass, int size) { int i = 0; int j = size - 1; int temp; int mid = mass[size / 2]; do { while (mass[i] < mid) { i++; } while (mass[j] > mid) { j--; } if (i <= j) { temp = mass[i]; mass[i] = mass[j]; mass[j] = temp; i++; j--; } } while (i <= j); if (j > 0) { sort3(mass, j + 1); } if (i < size) { sort3(&mass[i], size - i); } } </pre>	<pre> void sort4_prepare(int *mass, int size) { int middle, start_left, start_right, last; int *work_mas = new int[size]; middle = size / 2; start_left = 0; start_right = middle; last = size; for (int j = 0; j < last; j++) { if ((start_left < middle) && ((start_right >= last) (mass[start_left] < mass[start_right]))) { work_mas[j] = mass[start_left]; start_left++; } else { work_mas[j] = mass[start_right]; start_right++; } } for (int j = 0; j < last; j++) mass[j] = work_mas[j]; delete[] work_mas; }; void sort4(int *mass, int size) { if (size > 1) { sort4(mass, size / 2); sort4(&mass[size / 2], size - size / 2); sort4_prepare(mass, size); } }; </pre>

Какая из них быстрее отсортирует по возрастанию массив из 1000 целых чисел, изначально отсортированных по убыванию?

Задача 4

Была получена выборка из базы, содержащей имена пользователей и первые 6 байт их хешированных паролей. Для хеширования применялась стандартная хеш-функция.

```

admin      5994471abb01
user       435c554a2e9c
operator   5e884898da28
guest      84983c60f7da

```

Стало известно, что у пользователя с именем *admin* пароль – 12345.

Какое значение хешированного пароля добавится в базу при создании пользователя *operator* с паролем *qwerty123*?

Задача 5

При копировании исходного кода программы произошла ошибка. Помогите определить, какое число может быть на месте **▲**, чтобы функция *function()* всегда корректно выполнялась, и в результате ее выполнения на экран выводилось слово «Hello».

Листинг программы приведен ниже.

Паскаль	Си
<pre> procedure function(); r:array [0..ord('D')+ +ord('C')-ord(▲)] of char; begin for i:=ord('#')-ord('#') to ord('B')-ord('A')+ord(▲) do r[i]:=chr((ord('+')- ord(''))* i); if (ord(r[▲]) + ord(r[ord('-')- ord('+')]) + ord(r[ord('=')- ord('<')]) + ord(r[ord('>')- ord('>')]) = 12) then writeln('Hello') else writeln('Goodbye'); end; </pre>	<pre> void function() { char r['D'-'C' + ▲]; for (i='#'-'#'; i<('B'-'A' + ▲); i++) { r[i]=(char)((+'-')*i); } if (r[▲] + r['-'-'+'] + + r['='-'<'] + r['>'-'>']) == 12) { printf("Hello\n"); } else { printf("Goodbye\n"); } return; } </pre>