

Условия задач отборочного этапа

11 КЛАССЫ

Задача 1. Ошибка в коде.

В ходе разработки программного проекта произошел сбой в системе контроля версий. В результате в системе остались три версии одной и той же подпрограммы, любая из которых может оказаться верной. Многоточие «...» обозначает некоторый код, который заведомо верен. Необходимо определить, какая из версий верна.

1	<pre>char *SomeAlgorithm(char S[35]) { char *S1 = NULL, *S2 = NULL; S1 = (char *)malloc(35); S2 = (char *)malloc(35); strcpy(S2, S); ... strcpy(S1, S2); return S1; }</pre>
2	<pre>static char *SomeAlgorithm(char S[35]) { char *S1 = NULL, *S2 = NULL; S1 = (char *)malloc(35); S2 = (char *)malloc(35); strcpy(S2, S); ... strcpy(S1, S2); free(S2); return S1; }</pre>

3	<pre> char const *SomeAlgorithm(char S[35]) { char *S1 = NULL, *S2 = NULL; S1 = (char *)malloc(35); S2 = (char *)malloc(35); strcpy(S2, S); ... strcpy(S1, S2); S2 = free(35); return S1; } </pre>
---	---

Ответ: Верна копия 2.

Задача 2. Регулярное выражение.

Регулярные выражения предоставляют возможности для описания подстрок определенного вида. Правила записи регулярных выражений приведены в таблице.

В потоке текстовых данных встречаются числа, которые могут быть записаны в следующих форматах:

,56 1,1 1,1234 0,123 0,123E4 1,56E+12 31,4E-1

Общее описание таких форматов имеет вид:

*послед-цифр*_{opt} , *послед-цифр эксп-множитель*_{opt}

где *эксп-множитель* – это **Е***знак*_{opt}*послед-цифр*, *знак* – это «+» или «-», подпись *opt* – обозначает необязательность соответствующего элемента формата.

Какое регулярное выражение позволяет выделять числа всех указанных форматов.

Название элемента регулярного выражения	Пример элемента регулярного выражения	Описание
символ	a	Элемент представляет собой именно тот символ, который распознается в анализируемой строке, т.е. в данном случае это литера «a»
любой символ	.	Элемент обозначает произвольный символ, который распознается в анализируемой строке
пробельный символ	\s	Элемент обозначает символ пробела
диапазон символов, []	[abc] – любая из букв a, b, c [a-z0-9] – любая из малых букв латинского алфавита и цифра	В скобках перечисляется множество или диапазон упорядоченных символов
отрицание диапазона	[^a-z5] – ни маленькая буква латинского алфавита, ни цифра 5	Символы после знака ^ не должны входить в распознаваемую подстроку символов

указание количества вхождений символов	?	Означает ноль или одно вхождение символа в распознаваемую подстроку символов
	+	Означает одно или более вхождение символа в распознаваемую подстроку символов, причем берется как можно больше количество символов
	+?	Означает одно или более вхождение символа в распознаваемую подстроку символов, причем берется как можно меньше количество символов
	*	Означает ноль или более вхождение символа в распознаваемую подстроку символов, причем берется как можно больше количество символов «a*» – любое количество идущих подряд символов «a»
	*?	Означает ноль или более вхождение символа в распознаваемую подстроку символов, причем берется как можно меньше количество символов
	{n}	Означает n вхождений символа в распознаваемую подстроку символов
	{n, m}	Означает от n до m вхождений символа в распознаваемую подстроку символов
Извлечения подстроки	используются круглые скобки	Выражение: $([0-9]^+)s+([0-9]^+)$ извлекает из строки «мои числа 45 567 234 56» числа: «45» и «567»

Ответ: $(f[0-9]^+)s+=s+([0-9]^*,[0-9]^+E?[+-]?[0-9]^*)$

Задача 3. Подбор пароля.

Вводимые пользователями пароли преобразуются с помощью функции, исходный код которой представлен ниже. Возвращаемое функцией значение сверяется с эталоном, хранящимся в базе данных.

Известно, что для некоторого логина установлен пароль «OLYMPIAD». Предложите такой шаблон пароля из трёх символов, что он также пройдет проверку.

Комментарий. В шаблоне знак вопроса «?» обозначает произвольный символ. Например, шаблон «X?X», означает, что на первом и последнем

месте пароля должен находиться символ X, а второй символ может быть выбран произвольно.

Вычисление хэш-функции на языке Pascal

```
function Hash(pwd: string): integer;
var
  i, H: integer;
  temp: array[1..4] of char;
  p: ^char;
begin
  i := 1;
  p := addr(temp[1]);
  while (i <= min(length(pwd), 8)) do begin
    if (i mod 2 = 1) then begin
      p^ := pwd[i];
      p := p + 1;
    end;
    i := i + 1;
  end;
  while ((i <= 8) and (p <= addr(temp[4]))) do begin
    p^ := '@';
    p := p + 1;
    i := i + 1;
  end;
  H := ((ord(temp[1]) + ord(temp[4])) and 255) * 256;
  H := H + ((ord(temp[2]) + ord(temp[3])) and 255);
end;
```

Вычисление хэш-функции на языке C

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int Hash(const char *pwd) {
  int i, H;
  char temp[4], *p;
  p = &temp[0];
  for( i = 0; i < __min(strlen( pwd ), 8); i++)
    if( i % 2 == 0 )
      *p++ = pwd[i];

  for(; i < 8 && p <= &temp[3]; i++)
    *p++ = '@';

  H = ((temp[0] + temp[3]) & 255) * 256;
  H += (temp[1] + temp[2]) & 255;
  return H;
}
```

Ответ: P?i

Задача 4. Сеть.

Локальная сеть предприятия расположена в одноэтажном здании, представляющем собой 2014 комнат, разделенных коридором, проходящим через все здание. В каждой комнате размещено по одному компьютеру. Компьютер в комнате с номером N соединен отдельными кабелями с компьютерами в комнатах N-1 и N+1, где $2 \leq N \leq 2013$. Кроме того,

соединены кабелем и компьютеры в комнатах 1 и 2014. Кабели могут быть проложены как через коридор, так и сквозь стену, соединяющую соседние кабинеты. Про подобные локальные сети говорят, что они имеют логическую топологию «кольцо».

Конкуренты предприятия решили заняться техническим шпионажем. При этом у них имеется доступ только к коммуникациям, расположенным в коридоре.

В каком из вариантов нумерации кабинетов конкурентам удастся перехватывать весь трафик локальной сети предприятия?

Ответ: Традиционно: размещение четных и нечетных комнат по разные стороны центрального коридора.

Задача 5. Арифметика больших чисел.

Вычислите последнюю (младшую) цифру числа $2017^{(2016^{2015})}$.

Ответ:1.