

Межрегиональная предметная олимпиада КФУ по информатике
2012-2013 учебный год
9 класс

1. (15) Многочлен $P(x) = p_0 + p_1x + \dots + p_nx^n$ задаётся массивом своих коэффициентов. Написать программу для вычисления коэффициентов многочлена $Q(x) = P(x - a)$, где a, p_0, p_1, \dots, p_n - заданные вещественные числа, n - заданное натуральное число от 1 до 1000.

2. (25) К Универсиаде-2013 в Казани построили N спортивных и культурных объектов, которые пронумерованы натуральными числами. Между некоторыми из них спроектированы дороги с односторонним движением. Написать программу, которая определит, есть ли среди объектов такой, что отправившись от него можно вернуться к нему же, посетив другие объекты? Если существует хотя бы один такой объект, то напечатайте слово «ДА». В противном случае – слово «НЕТ».

Входные данные. В первой строке записаны числа N и M , количество объектов и число дорог в проекте ($1 \leq N, M \leq 1000$). В следующих M строках записаны пары номеров объектов, которые соединяются дорогами. Номера объектов – различные натуральные числа от 1 до 2000000000.

Пример.

Ввод	Вывод
4 4 1 2 2 7 7 5 5 2	ДА

Примечание. Таким объектом может быть, например, объект номер 2. Из объекта 2 можно попасть в объект 7, из объекта 7 можно попасть в объект 5, из объекта 5 можно попасть в объект 2.

3. (15) Числа Фибоначчи определяются следующим образом:

$$f(n) = \begin{cases} 0, & \text{если } n = 0 \\ 1, & \text{если } n = 1 \\ f(n-1) + f(n-2), & \text{если } n > 1 \end{cases}$$

Для вычисления числа Фибоначчи по его номеру была написана такая функция:

Pascal	C/C++
<pre>Function Fib (n : integer) : integer; Begin if n < 2 then Fib := n else Fib := Fib (n-1) + Fib (n-2) End;</pre>	<pre>int Fib (int n) { if (n < 2) return n; else return Fib (n-1) + Fib (n-2); }</pre>

Сколько раз эта функция вызывается для вычисления 7-го числа Фибоначчи с помощью вызова $Fib(7)$?

4. (25) Вся ось X окрашена в чёрный цвет. Несколько отрезков окрасили в белый цвет. (Отрезки могут пересекаться и вкладываться один в другой). Концы заданы целыми числами L (левый) и R (правый), $L < R$ (значения не превосходят 10^9 по абсолютной величине). Ось X получается разбитой на чередующиеся участки белого и чёрного цвета. Написать программу, которая найдёт самый длинный участок оси, который целиком окрашен в белый цвет и напечатает его длину.

Входные данные. Первая строка содержит натуральное число N – количество отрезков ($N < 1000$), следующие N строк содержат по 2 целых числа L_i, R_i – концы очередного отрезка.

Пример.

Ввод	Вывод
4 2 3 -2 0 1 3 2 4	3

5. (5) Сколько существует наборов значений логических переменных, при которых заданное условие является истинным:

$((a \rightarrow b) \rightarrow (b \rightarrow a)) \rightarrow ((b \rightarrow a) \rightarrow (a \rightarrow b)) ?$

Обозначения: \leftrightarrow эквивалентность, \rightarrow импликация (следование), \wedge конъюнкция (логическое умножение).

6. (15) На вход программе подаётся файл 'program.pas' ('program.cpp'), содержащий текст некоторой синтаксически правильной программы на языке Паскаль (Си). Ваша программа должна напечатать этот текст, удалив из него все примечания.

Пример.

Ввод	Вывод
Program sample; { пример } Var c : char; Begin Read (c); Write (' {=} ', c) End.	Program sample; Var x : char; Begin Read (x); Write (' {=} ', x) End.
# include <iostream> using namespace std; int main () /* sample */ { cin >> c; // input cout << " /* */" << c; return 0; }	# include <iostream> using namespace std; int main () { cin >> c; cout << " /* */" << c; return 0; }

**Межрегиональная предметная олимпиада КФУ по информатике
2012-2013 учебный год
10 класс**

1. (15) Написать программу для вычисления коэффициентов многочлена $P(x) = (x - a)^n(x - b)^m$, где a, b – заданные вещественные числа, n, m – заданные натуральные числа от 1 до 1000.

2. (25) К Универсиаде-2013 в Казани построили N спортивных и культурных объектов, которые пронумерованы натуральными числами. Между некоторыми из них спроектированы двусторонние дороги. По дорогам можно добраться от любого объекта до любого другого. Написать программу, которая найдёт и напечатает наименьшее число дорог, которые можно оставить в проекте, чтобы по ним по-прежнему можно было бы добраться от любого объекта до любого другого, и напечатает сами дороги в том же формате, что и при вводе.

Входные данные. В первой строке записаны числа N и M , количество объектов и число дорог в проекте ($N, M \leq 1000$). В следующих M строках записаны пары номеров объектов, которые соединяются дорогами. Номера объектов – различные натуральные числа от 1 до 2000000000.

Пример.

Ввод	Вывод
4 4	3
1 2	1 2
2 7	2 7
7 5	7 5
5 2	

Примечание. Можно оставить первые 3 дороги из примера.

3. (15) Число сочетаний из N по K ($N \geq K$) определяется следующим образом:

$$C_n^k = \begin{cases} 1, & \text{если } k = 0 \\ 1, & \text{если } n = k \\ C_{n-1}^k + C_{n-1}^{k-1}, & \text{иначе} \end{cases}$$

Для вычисления числа сочетаний была написана такая функция:

Pascal	C/C++
<pre>Function Combination (n, k : integer) : integer; Begin if (n = k) OR (k = 0) then Combination := 1 else Combination := Combination (n-1, k) + Combination (n-1, k-1) End;</pre>	<pre>int Combination (int n, int k) { if (n == k k == 0) return 1; else return Combination (n-1, k) + Combination (n-1, k-1); }</pre>

Сколько раз эта функция вызывается для вычисления результата при вызове `Combination (6, 3)` ?

4. (25) Вся ось X окрашена в чёрный цвет. Несколько отрезков окрасили в белый цвет. (Отрезки могут пересекаться и вкладываться один в другой). Концы заданы целыми числами L (левый) и R (правый), $L < R$ (значения не превосходят 10^9 по абсолютной величине). Ось X получается разбитой на чередующиеся участки белого и чёрного цвета. Написать программу, которая найдёт и напечатает число участков оси, которые целиком окрашены в белый цвет.

Входные данные. Первая строка содержит натуральное число N – количество отрезков ($N < 1000$), следующие N строк содержат по 2 целых числа L_i, R_i – концы очередного отрезка.

Пример.

Ввод	Вывод
4 2 3 -2 0 1 3 2 4	2

5. (5) Сколько существует наборов значений логических переменных, при которых заданное условие является истинным:

$((a \leftrightarrow b) \rightarrow (c \leftrightarrow d)) \wedge ((c \leftrightarrow d) \rightarrow (a \leftrightarrow b)) ?$

Обозначения: \leftrightarrow эквивалентность, \rightarrow импликация (следование), \wedge конъюнкция (логическое умножение).

6. (15) На вход программе подаётся файл 'program.pas' ('program.cpp'), содержащий текст некоторой синтаксически правильной программы на языке Паскаль (Си). Ваша программа должна напечатать этот текст, удалив из него все примечания.

Пример.

Ввод	Вывод
Program sample; { пример } Var c : char; Begin Read (c); Write (' {=} ', c) End.	Program sample; Var x : char; Begin Read (x); Write (' {=} ', x) End.
# include <iostream> using namespace std; int main () /* sample */ { cin >> c; // input cout << " /* */" << c; return 0; }	# include <iostream> using namespace std; int main () { cin >> c; cout << " /* */" << c; return 0; }

Межрегиональная предметная олимпиада КФУ по информатике
2012-2013 учебный год
11 класс

1. (15) Многочлен $P(x) = p_0 + p_1x + \dots + p_nx^n$ задаётся массивом своих коэффициентов. Написать программу для вычисления коэффициентов многочлена $Q(x) = (x-b)^m P(x-a)$, где $a, b, p_0, p_1, \dots, p_n$ - заданные вещественные числа, n, m - заданные натуральные число от 1 до 1000.

2. (25) К Универсиаде-2013 в Казани построили N спортивных и культурных объектов, между некоторыми из них есть дороги. Все дороги двусторонние. Написать программу, которая найдёт и напечатает наименьшее число дорог, которые нужно ещё построить, чтобы по ним можно было добраться от любого объекта до любого другого, и напечатает сами дороги в том же формате, что и при вводе.

Входные данные. В первой строке записаны числа N и M , количество объектов и число уже имеющихся дорог ($N, M \leq 1000$). В следующих M строках записаны пары номеров объектов, которые соединяются дорогами. Номера объектов - различные натуральные числа от 1 до 2000000000.

Пример.

Ввод	Вывод
4 3	2
1 2	2 3
3 4	4 5
5 6	

Примечание. Есть и другие варианты.

3. (15) Функция Аккермана определяется следующим образом:

$$f(a, b) = \begin{cases} b+1, & \text{если } a = 0 \\ f(a-1, 1), & \text{если } a > 0, b = 0 \\ f(a-1, f(a, b-1)), & \text{если } a > 0, b > 0 \end{cases}$$

Для её вычисления была написана такая функция:

Pascal	C/C++
<pre>Function Akkerman (n, k : integer) : integer; Begin if n = 0 then Akkerman := k+1 else if k = 0 then Akkerman := Akkerman (n-1, 1) else Akkerman := Akkerman (n-1, Akkerman (n, k-1)) End;</pre>	<pre>int Akkerman (int n, int k) { if (n == 0) return k+1; else if (k == 0) return Akkerman (n-1, 1); else return Akkerman (n-1, Akkerman (n, k-1)); }</pre>

Сколько раз эта функция вызывается для вычисления результата при вызове Akkerman (2, 3) ?

4. (25) Вся ось X окрашена в чёрный цвет. Несколько отрезков окрасили в белый цвет. (Отрезки могут пересекаться и вкладываться один в другой). Концы заданы целыми числами L (левый) и R (правый), $L < R$ (значения не превосходят 10^9 по абсолютной величине). Ось X получается разбитой на чередующиеся участки белого и чёрного цвета. Написать программу, которая найдёт и напечатает суммарную длину всех участков оси, которые целиком окрашены в белый цвет. Входные данные. Первая строка содержит натуральное число N – количество отрезков ($N < 1000$), следующие N строк содержат по 2 целых числа L_i, R_i – концы очередного отрезка.

Пример.

Ввод	Вывод
4 -2 0 2 3 1 3 2 4	5

5. (5) Сколько существует наборов значений логических переменных, при которых заданное условие является истинным:

$((a \rightarrow b) \leftrightarrow (b \rightarrow a)) \leftrightarrow ((b \rightarrow a) \leftrightarrow (a \rightarrow b))$?

Обозначения: \leftrightarrow эквивалентность, \rightarrow импликация (следование), \wedge конъюнкция (логическое умножение).

6. (15) На вход программе подаётся файл 'program.pas' ('program.cpp'), содержащий текст некоторой синтаксически правильной программы на языке Паскаль (Си). Ваша программа должна напечатать этот текст, удалив из него все примечания.

Пример.

Ввод	Вывод
Program sample; { пример } Var c : char; Begin Read (c); Write (' {=} ', c) End.	Program sample; Var x : char; Begin Read (x); Write (' {=} ', x) End.
# include <iostream> using namespace std; int main () /* sample */ { cin >> c; // input cout << " /* */" << c; return 0; }	# include <iostream> using namespace std; int main () { cin >> c; cout << " /* */" << c; return 0; }

Межрегиональная предметная олимпиада КФУ по информатике (решения) 2012-2013

9 класс

1. Реализовать процедуру умножения многочленов (точнее многочлена на двучлен $x-a$) и последовательно вычислять коэффициенты многочленов. Нужно учесть, что программа должна вычислять не значение многочлена в какой-то точке (значение x не задано!), а сам многочлен, т.е. его коэффициенты.

2. Если рассматривать объекты, как вершины графа, а дороги – как ориентированные рёбра, то задача сводится к нахождению любого цикла в ориентированном графе. Это можно сделать с помощью алгоритма DFS (обход в глубину, см. <http://e-maxx.ru/algo/dfs> или http://e-maxx.ru/algo/finding_cycle)

3. Последовательно вычисляем количество вызовов от малых значений аргумента: вычисление $F(0)$ и $F(1)$ не приводит к новым вызовам, для $F(2)$ – 2 новых вызова и т.д.

Другой способ - можно начать вычисления с построения дерева вызовов от первого вызова $F(7)$ и дальше вниз к $F(6)$ и $F(5)$ и т.д. Сам вызов $F(7)$ не учитываем.

Ответ: 40.

4. Упростить выражение до вида $a \rightarrow b$ и посчитать все наборы.

Можно просто перебрать все наборы и вычислить значение выражения (построить таблицу истинности).

Ответ: 3

10 класс

1. Реализовать процедуру умножения многочленов (точнее многочлена на двучлен $x-a$) и последовательно вычислять коэффициенты многочленов. Нужно учесть, что программа должна вычислять не значение многочлена в какой-то точке (значение x не задано!), а сам многочлен, т.е. его коэффициенты.

2. Если рассматривать объекты, как вершины графа, а дороги – как рёбра, то задача сводится к нахождению минимального остовного дерева в неориентированном графе, т. е. нужно избавиться от рёбер, которые образуют циклы в этом графе. Существует несколько алгоритмов решения этой задачи, например, алгоритм Прима или алгоритм Краскала (см. http://e-maxx.ru/algo/mst_prim или http://e-maxx.ru/algo/mst_kruskal).

3. Последовательно вычисляем количество вызовов от малых значений аргумента: вычисление $C(x,0)$ и $C(x,x)$ не приводит к новым вызовам, для $C(2,1)$ – 2 новых вызова и т.д.

Другой способ - можно начать вычисления количества вызовов с построения дерева вызовов от первого вызова $C(6,3)$ и дальше вниз к $C(6,2)$ и $C(5,3)$ и т.д. Сам вызов $C(6,3)$ не учитываем.

Ответ: 38.

4. Упростив выражение, можно увидеть, что пары a, b и c, d должны быть одновременно равны или одновременно не равны.

Можно просто перебрать все наборы и вычислить значение выражения (построить таблицу истинности).

Ответ: 8.

11 класс

1. Реализовать процедуру умножения многочленов (точнее многочлена на двучлен $x-a$) и последовательно вычислять коэффициенты многочленов. Нужно учесть, что программа должна вычислять не значение многочлена в какой-то точке (значение x не задано!), а сам многочлен, т.е. его коэффициенты.

2. Если рассматривать объекты, как вершины графа, а дороги – как рёбра, то задача сводится к нахождению компонент связности в заданном неориентированном графе. Последовательно находим компоненты и для каждой новой компоненты проводим новое ребро, соединяющее его с предыдущей компонентой (для этого ребра подойдут две любые вершины: одна из одной компоненты, другая из другой). Для поиска компонент связности можно использовать алгоритм BFS (обход в ширину, см. <http://e-maxx.ru/algo/bfs> или http://e-maxx.ru/algo/connected_components).

3. Первый способ состоит в вычислении количества вызовов для малых значений аргументов и дальше к большим. Например, вызов $A(0, x)$ не порождает новых вызовов, а вызов $A(1, 3)$ – 7 новых вызовов и т.д.

Другой способ - можно начать вычисления количества вызовов с построения дерева вызовов от первого вызова $A(2, 3)$ и дальше вниз к $A(1, A(2, 3))$, $A(1, A(1, A(2, 2)))$ и т.д. Сам вызов $A(2, 3)$ не учитываем.

Ответ 43.

4. Упростить выражение до вида $(a \leftrightarrow b) \leftrightarrow (b \leftrightarrow a)$.

Можно просто перебрать все наборы и вычислить значение выражения (построить таблицу истинности).

Ответ: 4.