

Вариант 6 (условия и решения)

Задача 1 (8 баллов). Перевести шестнадцатеричное число $A_{16} = F12,AE$ в десятичную систему счисления.

Решение задачи 1.

- 1) $F12_{16} = 15 \cdot 16^2 + 1 \cdot 16^1 + 2 \cdot 16^0 = 15 \cdot 256 + 1 \cdot 16 + 2 \cdot 1 = 3840 + 16 + 2 = 3858_{10}$.
- 2) $0,AE_{16} = 10 \cdot 16^{-1} + 14 \cdot 16^{-2} = 10/16 + 14/256 = 0,625 + 0,0546875 = 0,6796875_{10}$.

Ответ: $A_{10} = 3858,6796875$

Задача 2 (8 баллов). Найти сумму шестнадцатеричных чисел $A_{16} = C12F$ и $B_{16} = 359A$, используя 16-разрядный сумматор, старший разряд которого знаковый. Ответ дать в шестнадцатеричной форме. При переполнении разрядной сетки ответ сопроводить сообщением.

Решение задачи 2.

- 1) $[A_2]_{np} = 1,100\ 0001\ 0010\ 1111$ $[B_2]_{np} = 0,011\ 0101\ 1001\ 1010$
- 2) $[A_2]_d = 1,011\ 1110\ 1101\ 0001$ $[B_2]_d = 0,011\ 0101\ 1001\ 1010$
- 3) $[A_2]_d^M = 11,011\ 1110\ 1101\ 0001$ $[B_2]_d^M = 00,011\ 0101\ 1001\ 1010$
- 4) $[A_2]_d^M + [B_2]_d^M = 11,011\ 1110\ 1101\ 0001 + 00,011\ 0101\ 1001\ 1010 = 11,111\ 0100\ 0110\ 1011$
- 5) $[C_2]_{np} = 1,000\ 1011\ 1001\ 0101$

Ответ: $C_{16} = 8B95$

Задача 3 (8 баллов). Дано выражение, в котором используются операции над булевыми величинами, принимающими значения Т (истина) и F (ложь). Выражение может содержать круглые скобки и следующие знаки операций: отрицание (\neg), конъюнкция (\wedge), дизъюнкция (\vee), импликация (\rightarrow). Операции имеют следующие уровни приоритета: уровень 1 (\neg), уровень 2 (\wedge), уровень 3 (\vee), уровень 4 (\rightarrow). Построить таблицу истинности для выражения $((p \rightarrow q) \wedge (q \rightarrow \neg r)) \rightarrow (r \rightarrow p)$.

Решение задачи 3.

p	q	r	$p \rightarrow q$	$\neg r$	$q \rightarrow \neg r$	$(p \rightarrow q) \wedge (q \rightarrow \neg r)$	$r \rightarrow p$	$((p \rightarrow q) \wedge (q \rightarrow \neg r)) \rightarrow (r \rightarrow p)$
T	T	T	T	F	F	F	T	T
T	T	F	T	T	T	T	T	T
T	F	T	F	F	T	F	T	T
T	F	F	F	T	T	F	T	T
F	T	T	T	F	F	F	F	T
F	T	F	T	T	T	T	T	T
F	F	T	T	F	T	T	F	F
F	F	F	T	T	T	T	T	T

Задача 4 (8 баллов). Дано выражение, в котором используются поразрядные операции над 8-ми разрядными целыми числами без знака. В выражении используются круглые скобки и следующие знаки операций: поразрядное НЕ (\sim), поразрядное И ($\&$), поразрядное ИЛИ (\mid), поразрядный сдвиг влево (\ll), поразрядный сдвиг вправо (\gg). Операции имеют следующие уровни приоритета: уровень 1 (\sim), уровень 2 (\ll и \gg), уровень 3 ($\&$), уровень 4 (\mid). Вычислить значение следующего выражения: $(\sim a \mid a \ll 1 \& a \gg 1) \& ((a \mid b) \gg 1 \mid (a \& b) \ll 1)$ для $a = 195$ и $b = 60$. Ответ дать в двоичной и десятичной формах.

Решение задачи 4.

- 1) $a = 11000011_2$
- 2) $b = 00111100_2$
- 3) $\sim a = 3c_{16} = 00111100_2$
- 4) $a \ll 1 = 86_{16} = 10000110_2$
- 5) $a \gg 1 = 61_{16} = 01100001_2$
- 6) $a \ll 1 \& a \gg 1 = 0_{16} = 00000000_2$
- 7) $\sim a \mid a \ll 1 \& a \gg 1 = 3c_{16} = 00111100_2$
- 8) $a \mid b = ff_{16} = 11111111_2$
- 9) $(a \mid b) \gg 1 = 7f_{16} = 01111111_2$
- 10) $a \& b = 0_{16} = 00000000_2$
- 11) $(a \& b) \ll 1 = 0_{16} = 00000000_2$

$$12) (a | b) \gg 1 | (a \& b) \ll 1 = 7f_{16} = 01111111_2$$

$$13) (\sim a | a \ll 1 \& a \gg 1) \& ((a | b) \gg 1 | (a \& b) \ll 1) = 3c_{16} = 00111100_2$$

Ответ: $00111100_2 = 60_{10}$

Задача 5 (8 баллов). Пусть $\{a_n\}$ ($n \geq 1$) – последовательность, для которой $a_{n+2} - 4a_{n+1} + 3a_n = 0$. Вычислить a_{10} , зная, что $a_1 = 3, a_2 = 9$.

Решение задачи 5.

Первый способ. $a_1 = 3, a_2 = 9, a_3 = 27, a_4 = 81, a_5 = 243, a_6 = 729, a_7 = 2187, a_8 = 6561, a_9 = 19683, a_{10} = 59049$.

Второй способ. Можно найти выражения для a_n через n , считая, что a_1 и a_2 заданы. Общий член последовательности будет иметь вид: $a_n = 3^n$. Для $n = 10$ будем иметь: $a_{10} = 3^{10} = 59049$.

Ответ: $a_{10} = 59049$

Задача 6 (8 баллов). Дана инфиксная запись арифметического выражения $(a * (b + (d - (e + f)))) / ((g + h) * (i + j))$. Найти постфиксную запись этого выражения.

Решение задачи 6.

В соответствии с алгоритмом «сортировочной станции» Дейкстры будем иметь:

Стек	Вход	Выход
	$(a * (b + (d - (e + f)))) / ((g + h) * (i + j))$	
($a * (b + (d - (e + f)))) / ((g + h) * (i + j))$	
($* (b + (d - (e + f)))) / ((g + h) * (i + j))$	a
(*	$(b + (d - (e + f)))) / ((g + h) * (i + j))$	A
(* ($b + (d - (e + f)))) / ((g + h) * (i + j))$	A
(* ($+ (d - (e + f)))) / ((g + h) * (i + j))$	a b
(* (+	$(d - (e + f)))) / ((g + h) * (i + j))$	a b
(* (+ ($d - (e + f)))) / ((g + h) * (i + j))$	a b
(* (+ ($- (e + f)))) / ((g + h) * (i + j))$	a b d
(* (+ (-	$(e + f)))) / ((g + h) * (i + j))$	a b d
(* (+ (- ($e + f)))) / ((g + h) * (i + j))$	a b d
(* (+ (- ($+ f)))) / ((g + h) * (i + j))$	a b d e
(* (+ (- (+	$f)))) / ((g + h) * (i + j))$	a b d e
(* (+ (- (+	$)))) / ((g + h) * (i + j))$	a b d e f
(* (+ (-	$)))) / ((g + h) * (i + j))$	a b d e f +
(* (+	$)) / ((g + h) * (i + j))$	a b d e f + -
(*	$) / ((g + h) * (i + j))$	a b d e f + - +
	$/ ((g + h) * (i + j))$	a b d e f + - + *
/	$((g + h) * (i + j))$	a b d e f + - + *
/($(g + h) * (i + j))$	a b d e f + - + *
/(($g + h) * (i + j))$	a b d e f + - + *
/(($+ h) * (i + j))$	a b d e f + - + * g
/((+	$h) * (i + j))$	a b d e f + - + * g
/((+	$) * (i + j))$	a b d e f + - + * g h
/($* (i + j))$	a b d e f + - + * g h +
/(*	$(i + j))$	a b d e f + - + * g h +
/(* ($i + j))$	a b d e f + - + * g h +
/(* ($+ j))$	a b d e f + - + * g h + i
/(* (+	$j))$	a b d e f + - + * g h + i
/(* (+	$)$	a b d e f + - + * g h + i j
/(*	$)$	a b d e f + - + * g h + i j +
/		a b d e f + - + * g h + i j + *
		a b d e f + - + * g h + i j + * /

Ответ: $a b d e f + - + * g h + i j + * /$

Задача 7 (8 баллов). Определить количество способов соединения восьми точек на окружности четырьмя непересекающимися хордами.

Решение задачи 7.

Первый способ. Задачу можно решить с помощью метода полного перебора.

Второй способ. Задачу можно решить аналитически, т. к. рассматриваемая задача является частным случаем следующей задачи. На окружности расположены $2n$ точек. Сколькими способами эти точки попарно соединяются n хордами, не имеющими общих концов и непересекающимися. Пусть a_n — количество способов соединить $2n$ точек на окружности n непересекающимися хордами. Ясно, что $a_1 = 1$ и $a_2 = 2$. Можно показать, что $a_n = a_{n-1} + a_{n-2}a_1 + a_{n-3}a_2 + \dots + a_1a_{n-2} + a_{n-1}$. Фиксируем одну из данных $2n$ точек. Хорда, выходящая из неё, делит окружность на две дуги, причём на каждой дуге расположено чётное число данных точек. Если на одной дуге расположено $2k$ точек, то на другой — $2(n-k-1)$ точек; эти точки можно соединить непересекающимися хордами (не пересекающими первую хорду) $a_{n-k-1}a_k$ способами. Осталось просуммировать по k от 0 до $2n-2$. Таким образом, $a_3 = 2a_2 + a_1^2 = 5$, $a_4 = 2a_3 + 2a_1a_2 = 14$.

Третий способ. Можно показать, что число способов соединения восьми точек на окружности четырьмя непересекающимися хордами, определяется четвертым числом Каталана. Само число Каталана выражается формулой $C(n) = (2n)!/n!(n+1)!$. $C(4) = (2 \cdot 4)!/4!(4+1)! = (8)!/4!(5)! = (6 \cdot 7 \cdot 8)/(1 \cdot 2 \cdot 3 \cdot 4) = 7 \cdot 2 = 14$.

Ответ: 14

Задача 8 (12 баллов). Согласно опросу 250 телезрителей, 95 из них нравится смотреть новости, 125 предпочитают смотреть спорт, 125 – комедии, 25 – новости и комедии, 45 спорт и комедии, 35 – новости и спорт, 5 любят все три вида программ.

- Сколько телезрителей смотрят новости, но не смотрят спорт?
- Сколько телезрителей смотрят новости или спорт, но не любят комедии?
- Сколько телезрителей не любят смотреть ни новости, ни спорт?

Сколько телезрителей смотрят спорт и комедии, но не смотрят новости?

Решение задачи 8.

Задачу можно решить графически (с использованием диаграмм Эйлера-Венна). Предлагается аналитический метод (с использованием комбинаторного принципа сложения).

Пусть универсум U – множество из 250 телезрителей, N – множество телезрителей, которые смотрят новости, S – множество телезрителей, которые смотрят спорт, K – множество телезрителей, которые смотрят комедии. Множество телезрителей, которые смотрят новости, но не смотрят спорт, равно

$$|N| - |N \cap S| = 95 - 35 = 60.$$

Множество телезрителей, которые смотрят новости или спорт, но не любят комедии, равно

$$|N| + |S| - |N \cap S| - |N \cap K| - |S \cap K| + |N \cap S \cap K| = 95 + 125 - 35 - 45 - 25 + 5 = 120.$$

Множество телезрителей, которые не любят смотреть ни новости, ни спорт, состоит из множества телезрителей, которые смотрят только комедии или тех телезрителей, которые не смотрят ни одну из трех перечисленных программ. Множество телезрителей, которые смотрят хотя бы одну из трех перечисленных программ, равно

$$|N \cup S \cup K| = |N| + |S| + |K| - |N \cap S| - |S \cap K| - |K \cap N| + |N \cap S \cap K| = 95 + 125 + 125 - 35 - 45 - 25 + 5 = 245.$$

Множество телезрителей, которые не смотрят ни одну из трех перечисленных программ, равно

$$|U| - |N \cup S \cup K| = 250 - 245 = 5.$$

Тогда множество телезрителей, которые не смотрят ни новости, ни спорт, равно

$$|K| - |S \cap K| - |K \cap N| + |N \cap S \cap K| + (|U| - |N \cup S \cup K|) = 125 - 45 - 25 + 5 + 5 = 65.$$

Множество телезрителей, которые смотрят спорт и комедии, но не смотрят новости, равно

$$|S \cap K| - |N \cap S \cap K| = 45 - 5 = 40.$$

Ответ: а) 60; б) 120; в) 65; г) 40

Задача 9 (16 баллов). Дана последовательность целых положительных чисел. Требуется для каждого значения последовательности Y ($1 \leq Y \leq 10^{18}$) подсчитать количество различных положительных значений X таких, что $Y = X + \text{Rev}(X)$, где оператор Rev "разворачивает" число наоборот. Значение $\text{Rev}(X)$ не должно содержать ведущих нулей. Например, $\text{Rev}(350) = 53$ и $\text{Rev}(53) = 35$.

Входные данные. Входной файл содержит последовательность целых положительных чисел. Признак конца данных -1.

Выходные данные. В выходной файл для каждого значения входной последовательности вывести количество различных чисел, удовлетворяющих условию задачи.

Пример входных данных	Пример выходных данных
10	1
11	1
121	9
2999999999999999981	1
109	0
7087	0
59284	0
10201	1
-1	

Решение задачи 9.

Язык Си.

```
#include "stdafx.h"
#include <stdlib.h>
FILE *ifs, *ofs;
typedef long long int LLI;
LLI count(LLI v, LLI d, int isfirst) {
    if (d == 0 && v == 0) return 1;
    if (v < 0 || v >= d * 20) return 0;
    if (d == 1) { if ((v & 1) == 0) return 1; else return 0; }
    LLI r = 0;
    for (int p = v % 10; p < 20; p += 10) {
        r += ((10 - abs(p - 9)) - isfirst) * count( (v - p*d - p) / 10, d/100, 0 );
        isfirst = 0;
    }
    return r ;
}
LLI count(LLI v) {
    LLI r = 0;
    for (LLI d = 1; d <= v + 1; d *= 10) r += count( v, d, 1 );
    return r ;
}
int _tmain(int argc, _TCHAR* argv[]) {
    LLI number;
    if( ifs = fopen( "input.txt", "r" ) ) == NULL return 1;
    if( ofs = fopen( "output.txt", "w" ) ) == NULL return 2;
    while (true) {
        fscanf( ifs, "%lld", &number );
        if (number < 0) break;
        fprintf( ofs, "%lld\n", count( number ) );
    }
    fclose( ifs );
    fclose( ofs );
    return 0;
}
```

Задача 10 (16 баллов). На плоскости дан простой многоугольник (т. е. без самокасаний и самопересечений, но не обязательно выпуклый). Требуется решить вопрос о выпуклости многоугольника.

Входные данные. Входной файл содержит целое число N ($3 \leq N \leq 1000$) – количество вершин многоугольника и последовательность из N пар действительных координат вершин многоугольника. Все координаты по модулю не больше 10^6 . Вершины многоугольника заданы в порядке их обхода против часовой стрелки.

Выходные данные. В выходной файл вывести слово YES, если многоугольник выпуклый, и слово NO в противном случае.

Примеры входных данных	Примеры выходных данных
4 1 0 0 1 -1 0 0 -1	YES
4 0 0 5 0 1 1 0 5	NO

Решение задачи 10.

Язык Си.

```
#include "stdafx.h"
```

```

FILE *ifs, *ofs;
struct point { double x, y; };
int direction(struct point pi, struct point pj, struct point pk) {
    int result = 0;
    double temp = ((pk.x - pi.x) * (pj.y - pi.y) - (pj.x - pi.x) * (pk.y - pi.y));
    if (temp < 0) result = -1; else if (temp > 0) result = 1;
    return result;
}
int _tmain(int argc, _TCHAR* argv[]) {
    int n, k_pred, k_succ = 0;
    bool result = true;
    struct point a[1000];
    if( (ifs = fopen( "input.txt", "r" )) == NULL) return 1;
    if( (ofs = fopen( "output.txt", "w" )) == NULL) return 2;
    fscanf( ifs, "%d", &n );
    for (int i = 0; i < n; i++) fscanf( ifs, "%lf %lf", &a[i].x, &a[i].y );
    k_pred = direction( a[0], a[1], a[2] );
    for (int i = 1; i < n-2; i++) {
        k_succ = direction( a[i], a[i+1], a[i+2] );
        if ( k_pred * k_succ < 0 ) { result = false; break; } else k_pred = k_succ;
    }
    if ( result ) {
        k_succ = direction( a[n-2], a[n-1], a[0] );
        if ( k_pred * k_succ < 0 ) result = false; else k_pred = k_succ;
    }
    if ( result ) {
        k_succ = direction( a[n-1], a[0], a[1] );
        if ( k_pred * k_succ < 0 ) result = false;
    }
    fprintf( ofs, (result) ? "YES" : "NO" );
    fclose( ifs );
    fclose( ofs );
    return 0;
}

```

Язык Паскаль.

```

program Project_6_10;
{$APPTYPE CONSOLE}
uses SysUtils, Math;
type Point = record x, y: Double; end;
function Direction(pi, pj, pk: Point): TValueSign ;
begin
    Result := Sign((pk.x-pi.x)*(pj.y-pi.y)-(pj.x-pi.x)*(pk.y-pi.y));
end;
var n, k_pred, k_succ, i: Integer;
    result: Boolean;
    a: array[0..1000] of Point;
    f, g: TextFile;
begin
    AssignFile(f, 'input.txt');
    AssignFile(g, 'output.txt');
    try
        Reset(f);
        Rewrite(g);
        try
            result := true;
            k_succ := 0;
            Read(f, n);
            for i := 0 to n-1 do Read(f, a[i].x, a[i].y);
            k_pred := direction(a[0], a[1], a[2]);
            for i := 1 to n-3 do
                begin
                    k_succ := direction(a[i], a[i+1], a[i+2]);
                    if (k_pred * k_succ < 0) then
                        begin
                            result := false;
                            break;
                        end
                    else

```

```

        k_pred := k_succ;
    end;
    if result then
    begin
        k_succ := direction(a[n-2], a[n-1], a[0]);
        if (k_pred * k_succ < 0 ) then result := false
        else k_pred := k_succ;
    end;
    if result then
    begin
        k_succ := direction(a[n-1], a[0], a[1]);
        if (k_pred * k_succ < 0 ) then result := false;
    end;
    if result then Writeln(g, 'YES')
    else Writeln(g, 'NO');
    finally
        CloseFile(f);
        CloseFile(g);
    end;
except
    on EInOutError do Writeln('InOutError!');
end;
end.

```