— Общая часть —

1. Вычислить интеграл $\int\limits_0^{2\pi} \frac{dx}{2+\cos x}$.

▼ **Решение.** Найдем неопределенный интеграл:

$$\int \frac{dx}{2+\cos x} = \left\{ t = \operatorname{tg}\frac{x}{2} \right\} =$$

$$= 2\int \frac{dt}{3+t^2} = \frac{2}{\sqrt{3}} \operatorname{arctg}\frac{t}{\sqrt{3}} + C = \frac{2}{\sqrt{3}} \operatorname{arctg}\frac{\operatorname{tg}\frac{x}{2}}{\sqrt{3}} + C;$$

функция $\operatorname{tg}\frac{x}{2}$ разрывна в точке $x=\pi$. Одна из первообразных

$$F(x) = \begin{cases} \frac{2}{\sqrt{3}} \operatorname{arctg}\frac{\operatorname{tg}\frac{x}{2}}{\sqrt{3}}, \ x \in [0;\pi); \\ \frac{2}{\sqrt{3}} \operatorname{arctg}\frac{\operatorname{tg}\frac{x}{2}}{\sqrt{3}} + C^*, \ x \in (\pi; 2\pi]; \end{cases}$$

C^* находится их условия: $F(\pi-0) = F(\pi+0)$, тогда $C^* = \frac{2\pi}{\sqrt{3}}$.

▲

Ответ. $\frac{2\pi}{\sqrt{3}}$.

2. Решите уравнение $xy' - 2x^4y = 2y\ln y$.

▼ **Решение.** Используем замену неизвестной функции.

$$xy' - 2x^4y = 2y\ln y \Leftrightarrow xy^{-1}y' - 2\ln y = 2x^4 \Leftrightarrow \{z = \ln y\} \Leftrightarrow xz' - 2z = 2x^4.$$

Далее решим однородное уравнение $xz' - 2z = 0$.

$$\frac{dz}{z} = 2\frac{dx}{x} \Leftrightarrow z = \tilde{C}x^2.$$

Найдем $\tilde{C}$ методом вариации произвольной постоянной. Подставляя в неоднородное уравнение $z = \tilde{C}(x)x^2$, получим $\tilde{C}'(x) = 2x$, отсюда $\tilde{C}(x) = x^2 + C$, $z = Cx^2 + x^4$.

▲

Ответ. $y = e^{Cx^2+x^4}$.

3. Ане 15 лет, и в письме она использует смайлики. Каждый смайлик Ани состоит из двоеточия и произвольного конечного количества скобок одного типа: или открывающих круглых, или закрывающих. Например: «:)», «:)))))», «:((». Помимо смайликов, текст может содержать и обычные скобки. Если после смайлика нужно поставить скобку того же типа, то, чтобы не было путаницы, она ставится после пробела: «Привет!!!! (если помнишь меня:))))))».

Напишите функцию, которая будет определять разность между общим количеством скобок в «веселых» смайликах и суммой скобок в «грустных» смайликах Ани. Допускается только один проход по тексту письма и использование конечного объема памяти (копирование письма в переменную недопустимо). Функция должна быть написана на одном из языков: Pascal, C, C++. Выбор того, в каком виде представлено письмо, остается за вами.

▼ **Решение.** Наиболее простой способ решения — пройти по тексту письма, и при виде знака двоеточия перейти в режим 1, находясь в котором, функция считает скобочки. Если при проходе в режиме 1 встретилась не скобочка или скобочка не того типа, происходит переход обратно в режим 0, в котором функция пребывает по умолчанию. Важно отметить две вещи:

1. После смайлика может стоять скобка («Привет!! :)))(надеюсь ты меня не забыл)»), и она не должна засчитываться как относящаяся к смайлику, поэтому нужно запоминать предыдущую скобку.
2. Два смайлика могут идти подряд («А это я!!:)):))»), поэтому при переходе обратно в нулевой режим надо не забыть после этого проверить симол, не двоеточие ли это.

Эти две ситуации отражены в решении всех вариантов. Решения отличаются только действиями над счетчиком «внутри» смайлика.

Язык Pyhton

```
def count_smileys_1(s):
    mode = 0
    counter = 0
    prev_smiley = ''
    for x in s:
        if mode == 1:
            if x == ')' and prev_smiley != '(':
                counter += 1
                prev_smiley = ')'
            elif x == '(' and prev_smiley != ')':
                counter -= 1
                prev_smiley = '('
            else:
                mode = 0
        if x == ':':
            mode = 1
            prev_smiley = ''
    return counter
```

Язык C++

```
int count_smileys_1(char* s, int len_s) {
    int counter = 0;
    int mode = 0;
    char prev_smiley = '-';
    for(int i = 0; i < len_s; i++) {
        if(mode == 1) {
            if(s[i] == ')' && prev_smiley != '(') {
                counter++;
                prev_smiley = ')';
            }
            else if(s[i] == '(' && prev_smiley != ')') {
                counter--;
                prev_smiley = '(';
            }
            else {
                mode = 0;
            }
        }
        if(s[i] == ':') {
            mode = 1;
            prev_smiley = '-';
        }
    }
    return counter;
}
```

▲

Ответ. Код программы.

— ВМ —

4. Найдите температуру в бесконечном тонком однородном стержне с теплоизолированной поверхностью во все моменты времени $t > 0$, если материал стержня описывается коэффициентом $a^2 = 1$, плотность распределения его внутренних источников/поглотителей тепла задается функцией $e^{-3t}\sin x$, а температура при $t = 0$ равна $\cos 2x$.

▼ **Решение.** Математически задача записывается следующим образом:

$$\begin{cases} u_t = u_{xx} + e^{-3t}\sin x,\ u = u(x,t),\ x \in \mathbb{R},\ t > 0, \\ u(x,0) = \cos 2x,\ x \in \mathbb{R}. \end{cases}$$

Будем искать ее решение в виде $u = v + w$, где

$$\begin{cases} v_t = v_{xx} + e^{-3t}\sin x, \\ v(x,0) = 0. \end{cases} \qquad \begin{cases} w_t = w_{xx}, \\ w(x,0) = \cos 2x. \end{cases}$$

Решения этих задач будем искать в виде $v = T_1(t)\sin x$ и $w = T_2(t)\cos 2x$. Тогда

$$\begin{cases} T_1' = -T_1 + e^{-3t}, \\ T_1(0) = 0. \end{cases} \qquad \begin{cases} T_2' = -4T_2, \\ T_2(0) = 1. \end{cases}$$

Теорема единственности гарантирует отсутствие других решений.

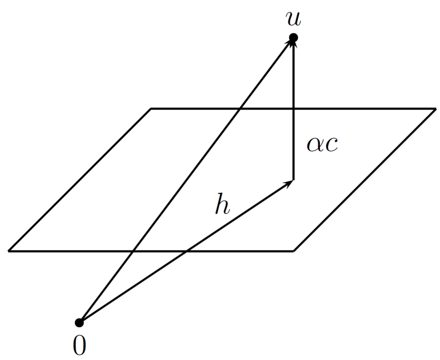
▲

Ответ. $\dfrac{e^{-t} - e^{-3t}}{2}\sin x + e^{-4t}\cos 2x.$

5. В пространстве $L_2[0, \pi/2]$ найти расстояние от точки $g(t) = t\cos t$ до множества

$$M = \left\{ f(t) \in L_2\left[0, \frac{\pi}{2}\right] :\ \int\limits_0^{\pi/2} 4\sin t f(t)dt = 0 \right\}.$$

▼ **Решение.** Заметим, что в задаче требуется в гильбертовом пространстве $\mathbb{H}$ найти расстояние от некоторой точкиu до множества M, которое представляет собой гиперплоскость, то есть $M = \{x \in \mathbb{H} :\ \langle c, x\rangle = 0\}$, где c – фиксированный элемент гильбертова пространства H.



Тогда имеет место следующее соотношение (см. рис.):

$$h = u - \alpha c, \tag{1}$$

где h – проекция точки u на множество M, c – вектор нормали гиперплоскости, α – некоторое число. Далее, так как $h \in M$, то

$$\langle c, h\rangle = 0 \tag{2}$$

Откуда с учетом (1) имеем

$$\alpha = \frac{\langle c, u\rangle}{\|c\|^2} \tag{3}$$

Таким образом проекция точки u на множество M равна

$$h = u - \frac{\langle c, u\rangle}{\|c\|^2}c, \tag{4}$$

а расстояние от точки u до множества M равно

$$\|\alpha c\| = \frac{|\langle c, u\rangle|}{\|c\|} \tag{5}$$

В варианте 1 $c = 4\sin t$, $u = t\cos t$

$$\|c\|^2 = \int\limits_0^{\pi/2} (4\sin t)^2 dt = \int\limits_0^{\pi/2} 8(1-\cos 2t)2dt = 4\pi,$$

$$\langle c, u\rangle = \int\limits_0^{\pi/2} t\cos t 4\sin t dt = \int\limits_0^{\pi/2} 2t\sin 2t dt = -\frac{1}{2}\int\limits_0^{\pi/2} 2t d\cos 2t = \frac{\pi}{2}.$$

И расстояние от точки до множества

$$\|\alpha c\| = \frac{\langle c, u\rangle}{\|c\|} = \frac{\sqrt{\pi}}{4}.$$

▲

Ответ. $\dfrac{\sqrt{\pi}}{4}$.

6. Построить монотонную разностную схему, аппроксимирующую со вторым порядком задачу

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = -f(x_1, x_2),\ (x_1, x_2) \in G = \{0 < x_1 < l_1,\ 0 < x_2 < l_2\};$$
$$\frac{\partial u}{\partial x_1}(0, x_2) = 0,\ 0 < x_2 < l_2;$$
$$u(x_1, x_2) = 0,\ (x_1, x_2) \in \partial G \backslash \{x_1 = 0,\ 0 < x_2 < l_2\}.$$

Выписать метод Якоби для решения разностных уравнений.

▼ **Решение.** Введем разностную сетку

$$x_{ij} = (x_{1,i}, x_{2,j});\ x_{1,i} = ih_1,\ x_{2,j} = jh_2;\ h_1N_1 = l_1,\ h_2N_2 = l_2;$$
$$\omega_{1h} = \{x_{ij};\ i = 1, 2, \dots, N_1 - 1;\ j = 1, 2, \dots, N_2 - 1\};$$
$$\omega_{2h} = \{x_{0j};\ j = 1, 2, \dots, N_2 - 1\};$$
$$\gamma_h = \{x_{i0}, x_{iN_2}, x_{N_1j};\ i = 0, 1, \dots, N_1 - 1;\ j = 1, 2, \dots, N_2 - 1\}.$$

Основное уравнение аппроксимируем со вторым порядком с использованием пятиточечного разностного оператора Лапласа:

$$y_{\bar{x}_1x_1,ij} + y_{\bar{x}_2x_2,ij} = -f_{ij},\ x_{ij} \in \omega_{1h}.$$

Рассмотрим аппроксимацию граничного условия 2-го рода на ω_{2h}.

$$u_{x_1,0j} = \frac{\partial u}{\partial x_1}(x_{0j}) + \frac{h_1}{2}\frac{\partial^2 u}{\partial x_1^2}(x_{0j}) + O(h_1^2) =$$
$$= \frac{\partial u}{\partial x_1}(x_{0j}) - \frac{h_1}{2}\left(\frac{\partial^2 u}{\partial x_2^2}(x_{0j}) + f(x_{0j})\right) + O(h_1^2) =$$
$$= \frac{\partial u}{\partial x_1}(x_{0j}) - \frac{h_1}{2}(u_{\bar{x}_2x_2,0j} + f_{0j}) + O(h_1^2 + h_2^2).$$

Приходим к уравнению

$$y_{x_1,0j} + 0.5h_1(y_{\bar{x}_2x_2,0j} + f_{0j}) = 0, \ x_{0j} \in \omega_2.$$

Учитывая равенства $y_{ij} = 0$, $x_{ij} \in \gamma_h$, монотонность схемы вытекает из канонической формы записи разностных уравнений:

$$\left(\frac{2}{h_1^2} + \frac{2}{h_2^2}\right) y_{ij} = \frac{1}{h_1^2}(y_{i-1j} + y_{i+1j}) + \frac{1}{h_2^2}(y_{ij-1} + y_{ij+1}) + f_{ij}, \ x_{ij} \in \omega_{1h};$$
$$\left(\frac{2}{h_1^2} + \frac{2}{h_2^2}\right) y_{0j} = \frac{2}{h_1^2}y_{1j} + \frac{1}{h_2^2}(y_{0j-1} + y_{0j+1}) + f_{0j}, \ x_{0j} \in \omega_{2h}.$$

▲

Ответ.

$$y_{\bar{x}_1x_1,ij} + y_{\bar{x}_2x_2,ij} = -f_{ij}, \ x_{ij} \in \omega_{1h};$$
$$y_{x_1,0j} = -0.5h_1(y_{\bar{x}_2x_2} + f)_{0j}, \ x_{0j} \in \omega_{2h};$$
$$y_{ij} = 0, \ x_{ij} \in \gamma_h.$$

Метод Якоби:

$$\frac{y_{i-1j}^{(n)} - 2y_{ij}^{(n+1)} + y_{i+1j}^{(n)}}{h_1^2} + \frac{y_{ij-1}^{(n)} - 2y_{ij}^{(n+1)} + y_{ij+1}^{(n)}}{h_2^2} = -f_{ij}, \ x_{ij} \in \omega_{1h};$$
$$\frac{-2y_{0j}^{(n+1)} + 2y_{1j}^{(n)}}{h_1^2} + \frac{y_{0j-1}^{(n)} - 2y_{0j}^{(n+1)} + y_{0j+1}^{(n)}}{h_2^2} = -f_{0j}, \ x_{0j} \in \omega_{2h};$$
$$y_{ij}^{(n+1)} = 0, \ x_{ij} \in \gamma_h.$$

— ММ —

4. Пусть ξ и η – независимые случайные величины, определенные на одном вероятностном пространстве. ξ имеет равномерное распределение на отрезке $[1,2]$, а η – дискретная случайная величина, принимающая значения -1, 0, 1 с вероятностью 1/3 каждое. Пусть $\zeta = e^{\xi\eta}$. Найти математическое ожидание случайной величины ζ.

▼ **Решение.** По формуле полной вероятности $E\zeta = E\left(E\left(\zeta|\eta\right)\right)$. Далее

$$E\left(E\left(\zeta|\eta\right)\right) = \frac{1}{3}\left(E\left(\zeta|\eta=-1\right) + E\left(\zeta|\eta=0\right) + E\left(\zeta|\eta=1\right)\right).$$

Поскольку ξ и η независимы

$$E\left(\zeta|\eta=-1\right) = Ee^{-\xi} = \int\limits_1^2 e^{-x}dx,$$

$$E\left(\zeta|\eta=0\right) = 1,$$

$$E\left(\zeta|\eta=1\right) = Ee^{\xi} = \int\limits_1^2 e^{x}dx.$$

Вычисляя интегралы, получаем ответ.

▲

Ответ. $E\zeta = \dfrac{1}{3}\left(-e^{-2} + e^{-1} + 1 - e + e^2\right).$

5. Найти точку минимума функции

$$f(x_1, x_2, x_3) = 8x_1x_2x_3 + 2/(x_1x_2) + 4/(x_1x_3) + 5/(x_2x_3)$$

в области $x_1 > 0, x_2 > 0, x_3 > 0$.

▼ **Решение.** Условия оптимальности первого порядка

$$\frac{\partial f}{\partial x_1} = 8x_2x_3 - \frac{2}{x_1^2x_2} - \frac{4}{x_1^2x_3} = 0,$$
$$\frac{\partial f}{\partial x_2} = 8x_1x_3 - \frac{2}{x_1x_2^2} - \frac{5}{x_2^2x_3} = 0,$$
$$\frac{\partial f}{\partial x_3} = 8x_1x_2 - \frac{4}{x_1x_3^2} - \frac{5}{x_2x_3^2} = 0.$$

Деля первое уравнение на x_2, а второе — на x_1, и, вычитая их, получим $x_2 = \frac{5}{4}x_1$. Аналогично находим, что $x_3 = \frac{5}{2}x_1$. Минимум функции

$$f\Big(x_1, \frac{5}{4}x_1, \frac{5}{2}x_1\Big) = 25x_1^3 + \frac{24}{5x_1^2}$$

достигается при $x_1 = (\frac{16}{125})^{1/5} = \frac{2}{5}(\frac{25}{2})^{1/5}$.
▲

Ответ. $x_1 = \frac{2}{5}\left(\frac{25}{2}\right)^{1/5}$, $x_2 = \frac{1}{2}\left(\frac{25}{2}\right)^{1/5}$, $x_3 = \left(\frac{25}{2}\right)^{1/5}$.

6. Найти схемную сложность булевой функции $f(\tilde{x}^2) = (0110)$ в элементарном базисе.

▼ **Решение.** Схема (формула) сложности четыре — $(x \vee y)\&\overline{x\&y}$. Докажем отсутствие схемы меньшей сложности. Схема меньшей сложности должна содержать два двухвходовых элемента и один элемент отрицания (кроме задачи варианта 3). Кроме этого пути от входов, соответствующих переменным, по которым функции немонотонны и не антимонотоны, должны проходить как через отрицание, так и мимо него. Выходной элемент не может быть отрицанием. Оставшиеся варианты схем нетрудно перебрать. Для медианы (задача варианта 3) сначала доказывается отсутствие отрицаний, а потом невозможность схем с двумя конъюнкциями и одной дизъюнкцией (в силу самодвойственности и наоборот).
▲

Ответ. 4 ($(x \vee y)\&\overline{x\&y}$).

— СП —

4. База данных компании «Рога и Копыта» хранит ф. и. о., должности, зарплаты сотрудников, проекты и задания. Предполагается, что должность сотрудника определяет его зарплату. Сотрудник может участвовать в нескольких проектах, но в каждом проекте выполняет только одно задание. Должность сотрудника определяется штатным расписанием компании и не зависит от проектов, в которых он занят.

Проектировщик схемы БД предложил всю информацию помещать в одно отношение (на данный момент он уже не работает в компании). Вот это отношение:

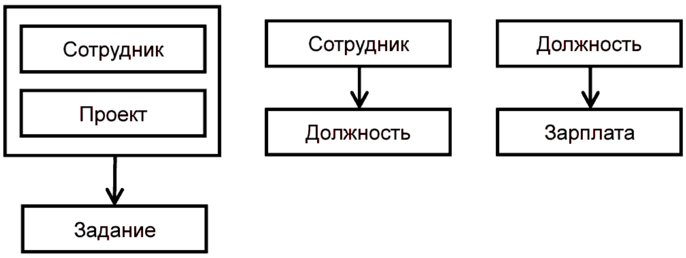
Сотрудник	*Должность*	*Зарплата*	*Проект*	*Задание*
Иванов А. Б.	руководитель	20000	Проект X	Ядро
Петров В. Г.	программист	15000	Проект X	Ядро
Сидоров Д. Е.	программист	15000	Проект X	Оболочка
Иванов А. Б.	руководитель	20000	Проект Y	Ядро
Петров В. Г.	программист	15000	Проект Y	Оболочка
Сидоров Д. Е.	программист	15000	Проект Y	Оболочка

Первичный ключ в этом отношении – {Сотрудник, Проект}. Предполагается, что ф. и. о. сотрудников компании не совпадают.

Диаграмма функциональных зависимостей выглядит следующим образом:

Требуется: 1) привести пример аномалии, возникающей при добавлении новых кортежей в исходное отношение; 2) используя теорему Хита, выполнить декомпозицию без потерь для исходного отношения так, чтобы все полученные отношения находились в третьей нормальной форме, и чтобы количество полученных отношений было минимальным. Полученные после декомпозиции отношения следует выписать в ответе.

▼ **Решение.** После декомпозиции получатся следующие функциональные зависимости:



Приводить схему в решении необязательно.
▲

Ответ. 1) Должна быть описана одна из указанных далее аномалий (или обе). а) Невозможно добавить нового сотрудника (указав должность и зарплату), который еще не участвует ни в одном проекте. б) Невозможно сохранить данные о новой должности (и соответствующей этой должности зарплате), пока не появится первый сотрудник с этой должностью.

2) Должны получиться 3 отношения:

Сотрудник	*Должность*
Иванов А. Б.	руководитель
Петров В. Г.	программист
Сидоров Д. Е.	программист

Должность	*Зарплата*
руководитель	20000
программист	15000

Сотрудник	*Проект*	*Задание*
Иванов А. Б.	Проект X	Ядро
Петров В. Г.	Проект X	Ядро
Сидоров Д. Е.	Проект X	Оболочка
Иванов А. Б.	Проект Y	Ядро
Петров В. Г.	Проект Y	Оболочка
Сидоров Д. Е.	Проект Y	Оболочка

5. Написать программу на языке Си для выполнения в операционной системе семейства Unix (считать, что подключения всех необходимых заголовочных файлов уже выполнены). В аргументах командной строки программе передаётся число N и имена текстовых файлов. Каждый текстовый файл должен содержать одну строку — имя исполняемого файла. Программа запускает на параллельное исполнение не более чем первые N исполняемых файлов и после окончания параллельного исполнения на последовательное исполнение — оставшиеся исполняемые файлы. По окончании работы процесс-родитель выводит на экран число — количество потомков, которые были успешно запущены и завершили свою работу с пользовательским кодом возврата 0.

▼ **Решение.**

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

```
#include <limits.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <sys/wait.h>

char *
getl(char *buf, size_t size, const char *path)
{
    FILE *f = fopen(path, "r");
    if (!f) return NULL;
    if (!fgets(buf, size, f)) {
        fclose(f);
        return NULL;
    }
    fclose(f); f = NULL;

    int len = strlen(buf);
    while (len > 0 && isspace(buf[len - 1])) --len;
    buf[len] = 0;
    return buf;
}

int
main(int argc, char *argv[])
{
    int n;
    sscanf(argv[1], "%d", &n);
    int arg = 2;
    int success_count = 0;
    int r;
    for (; arg < argc && arg < n + 2; ++arg) {
        if (!fork()) {
            char buf[PATH_MAX];
            if (getl(buf, sizeof(buf), argv[arg])) {
                execlp(buf, buf, NULL);
                exit(1);
            }
        }
    }
    while (wait(&r) > 0) {
        success_count += (WIFEXITED(r) && !WEXITSTATUS(r));
    }
    for (; arg < argc; ++arg) {
        if (!(r = fork())) {
            char buf[PATH_MAX];
            if (getl(buf, sizeof(buf), argv[arg])) {
                execlp(buf, buf, NULL);
                exit(1);
            }
        } else if (r > 0) {
            wait(&r);
            success_count += (WIFEXITED(r) && !WEXITSTATUS(r));
        }
    }
```

```
    printf("%d\n", success_count);
    return 0;
}
```

▲

Ответ. Код программы.

6. Определить последовательную и параллельную сложность алгоритма, записанного с помощью следующего фрагмента программы:

```
for( i = 1 ; i <= n ; ++i)
 for( j = 1 ; j <= m ; ++j)
  A[i][j] = A[i][j] * A[i-1][j] * A[i][j-1] ;
```

где: последовательная сложность — это число операций умножения в теле цикла; параллельная сложность — это длина критического пути графа алгоритма (число вершин в критическом пути) данного фрагмента, где каждая вершина представляет отдельное срабатывание оператора тела цикла. Ответ обосновать. Без обоснования ответ не засчитывается.

▼ **Решение.** Для определения последовательной сложности нужно определить общее число операций: тело данного цикла будет выполнено n*m раз, причем каждому выполнению оператора, стоящему в теле цикла, соответствует две операции умножения, а значит и общее число операций, и последовательная сложность равны $2 * n * m$. Для определения параллельной сложности алгоритма необходимо построить граф алгоритма, для данного фрагмента он выглядит следующим образом (см. ответ). Очевидно, что параллельная сложность, т.е. длина критического пути данного графа (число вершин в критическом пути) равна $n + m - 1$.
▲

Ответ. Последовательная сложность $= 2 * n * m$ операций. Параллельная сложность $= n + m - 1$ шагов.