

БИЛЕТ 1 — Часть Б
Зад.2: Обратное проектирование mystery(lst)
1. Фильтрует список — оставляет только числа > 0
2. Вход: lst (список) -> Выход: list (только >0)
3. Блок-схема: для x in lst -> x>0? -> Да:append(x) -> return result

БИЛЕТ 3 — Часть Б
Зад.2: Тест-кейсы get_stipend(score)
1.9->Ошибка 2.0->0 2.9->0 3.0->2000
3.9->2000 4.0->5000 5.0->10000 5.1->Ошибка

БИЛЕТ 5 — Часть Б
Зад.2: Первое отрицательное + Чепин
<pre>def first_neg(lst): for x in lst: if x < 0: return x return None</pre>
Чепин: P=1(lst), M=1(x), C=1(if) -> Q=1+2+3=6

БИЛЕТ 7 — Часть Б
Зад.2: Фильтрация положительных + Холстед
<pre>def filter_pos(lst): return [x for x in lst if x > 0]</pre>
n1=5, n2=4, N1=7, N2=8 -> V=15xlog2(9)≈47.5

БИЛЕТ 9 — Часть Б
Зад.2: Тест-кейсы check_password(pwd)
Кл.1: длина <8 ("abc1") -> Невалидный
Кл.2: 8-16 + цифра ("abcdef1g") -> Валидный
Кл.3: 8-16, без цифры ("abcdefgh") -> Невалидный
Кл.4: длина >16 -> Невалидный

БИЛЕТ 11 — Часть Б
Зад.2: Чётность числа + Чепин
<pre>def is_even(n): if n%2==0: return True return False</pre>
Чепин: P=1(n), M=0, C=1(if) -> Q=1+0+3=4

БИЛЕТ 13 — Часть Б
Зад.2: if-elif оценки + Чепин
<pre>def get_grade(s): if s>=90: return 5 elif s>=75: return 4 elif s>=60: return 3 else: return 2</pre>
Чепин: P=1(s), M=0, C=3(if/elif/elif) -> Q=1+0+9=10

БИЛЕТ 15 — Часть Б
Зад.3: Тест-кейсы delivery_cost(weight)
0.5->200 1.0->200 1.01->350
5.0->350 5.01->600 10.0->600

БИЛЕТ 17 — Часть Б
Зад.2: Чётность + Чепин (минимальный)
<pre>def is_even(n): return n % 2 == 0</pre>
P=1(n), M=0, C=0, T=0 -> Q=1

БИЛЕТ 19 — Часть Б
Зад.2: Обратное проектирование mystery(n)
1. Вычисляет n! (факториал)
2. Вход: int n -> Выход: int
3. result=1 -> n>1? -> Да:result*=n,n-=1 -> return result

БИЛЕТ 21 — Часть Б
Зад.2: Максимум двух чисел + Чепин
<pre>def max_two(a, b): if a>b: return a return b</pre>
Чепин: P=2(a,b), M=0, C=1(if) -> Q=2+0+3=5

БИЛЕТ 2 — Часть Б
Зад.2: Рефакторинг price1/price2
<pre>def apply_discount(price: float) -> float: """"Скидка 10%, возвращает итог.""" return price * 0.9</pre>
Устранено дублирование, убран print из логики.

БИЛЕТ 4 — Часть Б
Зад.2: Сумма квадратов чётных + Чепин
<pre>def sum_even_sq(lst): return sum(x**2 for x in lst if x%2==0)</pre>
Чепин: P=1(lst), M=0, C=1(if) -> Q=1+0+3=4

БИЛЕТ 6 — Часть Б
Зад.2: Сумма 1..N + Чепин
<pre>def sum_to_n(n): s,i=0,1 while i<=n: s+=i; i+=1 return s</pre>
Чепин: P=1(n), M=2(s,i), C=1(while) -> Q=1+4+3=8

БИЛЕТ 8 — Часть Б
Зад.2: Рефакторинг f/g
<pre>def power_sum(x: float) -> float: """"Квадрат + куб числа x.""" return x**2 + x**3</pre>
Проблемы: дублирование, имена f/g/y/z неинформативны

БИЛЕТ 10 — Часть Б
Зад.2: Сумма/среднее/кол-во + Чепин
<pre>def stats(lst): n=len(lst); s=sum(lst) return s, s/n, n</pre>
Чепин: P=1(lst), M=2(s,n), C=0 -> Q=1+4=5

БИЛЕТ 12 — Часть Б
Зад.2: Факториал while + Холстед
<pre>def factorial(n): r=1 while n>1: r*=n; n-=1 return r</pre>
n1=6, n2=3, N1=8, N2=7 -> V=15xlog2(9)≈47.5

БИЛЕТ 14 — Часть Б
Зад.2: Тест-кейсы fizzbuzz(n)
n=0->0 n=3->"Fizz" n=5->"Buzz"
n=15->"FizzBuzz" n=-1->-1

БИЛЕТ 16 — Часть Б
Зад.2: Знак числа + Чепин
<pre>def sign(n): if n>0: return "положительное" elif n<0: return "отрицательное" else: return "ноль"</pre>
Чепин: P=1(n), M=0, C=2(if/elif) -> Q=1+0+6=7

БИЛЕТ 18 — Часть Б
Зад.2: Макс. чётное + Чепин
<pre>def max_even(lst): r=None for x in lst: if x%2==0 and (r is None or x>r): r=x return r</pre>
Чепин: P=1, M=2(r,x), C=2(if) -> Q=1+4+6=11

БИЛЕТ 20 — Часть Б
Зад.2: Голосование + Чепин
<pre>def can_vote(age): return age >= 18</pre>
Чепин: P=1(age), M=0, C=1(>=) -> Q=1+0+3=4

БИЛЕТ 22 — Часть Б
Зад.2: Угадай число + Холстед
<pre>import random def guess_game(): n=random.randint(1,10) while True: g=int(input("Число: ")) if g==n: print("Верно!"); break elif g>n: print("Меньше") else: print("Больше")</pre>
n1≈8,n2≈6,N1≈12,N2≈10 -> V≈22xlog2(14)≈84

БИЛЕТ 23 — Часть Б
Зад.2: Простое число + Холстед
<pre>def is_prime(n): if n<2: return False for i in range(2,int(n**0.5)+1): if n%i==0: return False return True</pre>
n1=7, n2=5, N1=10, N2=9 -> V=19xlog2(12)≈68

БИЛЕТ 25 — Часть Б
Зад.2: Обратное проектирование process(data)
1. Берёт из списка int>0 и возводит в квадрат
2. Вход: list(смешанный) -> Выход: list[int]
3. для item: isinstance(int) И >0? -> append(item^2)

БИЛЕТ 27 — Часть Б
Зад.2: Делители числа + Чепин
<pre>def get_divisors(n): return [i for i in range(1,n+1) if n%i==0]</pre>
Чепин: P=1(n), M=1(i), C=1(if) -> Q=1+2+3=6

БИЛЕТ 29 — Часть Б
Зад.3: Документирование + рефакторинг is_adult
<pre>def is_adult(age: int) -> bool: """ Args: age - возраст Returns: True если age>=18 """ return age >= 18</pre>

БИЛЕТ 31 — Часть Б
Зад.2: Итоговая оценка + Чепин
<pre>def final_grade(grades): avg=sum(grades)/len(grades) if avg>=4.5: return 5 elif avg>=3.5: return 4 elif avg>=2.5: return 3 else: return 2</pre>
Чепин: P=1, M=1(avg), C=3 -> Q=1+2+9=12

БИЛЕТ 33 — Часть Б
Зад.2: Ошибки в Game()
1. g=input() — строка, нужен int(input())
2. Три отдельных if вместо if/elif
3. Рекурсия Game() без выхода -> стек переполнен
<pre>def game(): n=random.randint(1,10) g=int(input("Попытка: ")) if g==n: print("Угадал!") elif g>n: print("Много") else: print("Мало")</pre>

БИЛЕТ 35 — Часть Б
Зад.3: Ошибки в calc_avg
Ош.1: calc_avg([]) -> ZeroDivisionError
Ош.2: calc_avg(None) -> TypeError
<pre>def calc_avg(nums): if not nums: return 0 return sum(nums)/len(nums)</pre>

БИЛЕТ 37 — Часть Б
Зад.2: Ошибки в safe_divide
Ош.1: нет обработки деления на 0
Ош.2: нет проверки типов -> TypeError при "5"/2
<pre>def safe_divide(a,b): if not isinstance(a,(int,float)) or not isinstance(b,(int,float)): raise TypeError("Нужны числа") if b==0: raise ZeroDivisionError() return a/b</pre>

БИЛЕТ 39 — Часть Б
Зад.2: Тест-кейсы get_grade(score)
59->"2" 60->"3" 74->"3" 75->"4"
89->"4" 90->"5" 100->"5"

БИЛЕТ 24 — Часть Б
Зад.2: Тип треугольника + Холстед n1/n2
<pre>def tri_type(a,b,c): if a==b==c: return "равносторонний" elif a==b or b==c or a==c: return "равнобедренный" else: return "разносторонний"</pre>
n1=7(def,if,elif,else,==,or,return) n2=6(a,b,c+3 строки)

БИЛЕТ 26 — Часть Б
Зад.2: Ревью login — проблемы
1. Пароли в открытом виде в коде (безопасность)
2. users[username] -> KeyError если нет юзера
3. Нет try/except
<pre>def login(u,p): users={"admin":"hash"} return users.get(u)==p</pre>

БИЛЕТ 28 — Часть Б
Зад.2: Классификация возраста + Чепин
<pre>def classify_age(age): if age<=12: return "ребёнок" elif age<=17: return "подросток" elif age<=64: return "взрослый" else: return "пожилой"</pre>
Чепин: P=1(age), M=0, C=3 -> Q=1+0+9=10

БИЛЕТ 30 — Часть Б
Зад.2: Ошибки в check_pass
Ош.1(логика): p=="admin" or "pass123" — всегда True!
Ош.2(стиль): p — неинформативное имя
<pre>def check_pass(): pw=input("Пароль: ") if pw=="admin" or pw=="pass123": print("Доступ!"); return True</pre>

БИЛЕТ 32 — Часть Б
Зад.2: Сравнение чисел + Холстед
<pre>def compare(a,b): if a>b: return "а больше" elif a<b: return "b больше" else: return "равны"</pre>
n1=7, n2=5, N1=8, N2=7 -> V=15xlog2(12)≈53.8

БИЛЕТ 34 — Часть Б
Зад.2: Ошибки в classify_score
Ош.1: score=input() — перезаписывает параметр (строка!)
Ош.2: сравнение строки с числом -> TypeError
<pre>def classify_score(name:str, score:int): if score>100: grade="Отлично" elif score>70: grade="Хорошо" else: grade="Плохо" print(name+" получил: "+grade)</pre>

БИЛЕТ 36 — Часть Б
Зад.2: Обратное проектирование f(n) — сумма 1..n
1. Сумма чисел от 1 до n включительно
2. Имя: sum_to_n(n)
3. s=0,i=1 -> i<=n? -> s+=i,i+=1 -> return s

БИЛЕТ 38 — Часть Б
Зад.2: Рефакторинг калькулятора
<pre>def calculate(a:float, b:float, op:str)->float: """Арифметика. op: +,-,*,/""" ops={"+":a+b,"-":a-b,"*":a*b} if op=="/": if b==0: raise ZeroDivisionError() return a/b if op not in ops: raise ValueError(f"Неизвестный: {op}") return ops[op]</pre>

БИЛЕТ 40 — Часть Б
Зад.3: Документирование calc_discount
<pre>def calc_discount(price:float, promo:str)->float: """ Args: price-цена; promo-промокод Returns: цена после скидки Example: calc_discount(100,"SALE")->80.0 """ return price*0.8 if promo=="SALE" else price</pre>