



Вариант № 2

Оценка выполнения олимпиадной работы (заполняется проверяющим)											
№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Полученный балл	10	-	1	10	10	20					51
I проверка	Подпись проверяющего			Σ баллов	сто тысяч один						
	Подпись проверяющего			Σ баллов							
II проверка	Подпись проверяющего			Σ баллов	сто тысяч один						
	Подпись проверяющего			Σ баллов							



Вариант № 2

Набравшие баллы	Твердо когда	Друге иногда	Масса- масса	Средняя масса
Сила	2000	128100	121 +	255255 2480
Полностью или не	1000	128180	122 +	255255 2520
Математическое определение	500	1281120	123 +	2552540
Легкость	30	1281440	126 +	255255 255192
Средняя длина	14	12811484	127 +	255255 255224
Полностью	4	12811496	129 +	255255 255248



Вариант № 2

4 (продолжение)

Название подсети	Интервал по указаниям	Диапазон доступности адресов для ангелов
Филчан	128101 T	128101 - 12817254
Производительность корпуса	128181 T	128181 - 128111254
Науч-порт ангелов	1281121 1281121 +	1281121 - 128173254
Лаборатория	1281141 T	1281141 - 12811462
Служба безопасности	12811465 +	12811465 - 12811494
IT-ангел	12811497 T	12811497 - 128114102

105-?

(10)

исходная сеть не пробивалась.



Вариант № 2

✓ 6

```
import sys
directions = [(-1, 0), (0, 1), (1, 0), (0, -1)]
def is_valid(grid, i, j): - проверим границы и что
    return 0 <= i and i < len(grid) and
    0 <= j and j < len(grid[i]) and grid[i][j] ==
def next_cell(grid, d, i, j):
    for rot in range(3):
        delta = directions[d]
        ni, nj = i + delta[0], i + delta[1]
        if is_valid(grid, ni, nj):
            return d, ni, nj
    d = (d + 1) % 4
    return -1, -1, -1
```



Вариант № _____

№6 (адаптировано)

```
def launch (grid, d, i, j):
    cells = set()
    states = set()
    while (d, i, j) not in states and is
    is valid (grid, i, j):
        states.add((d, i, j))
        cells.add((i, j))
        d, i, j = next cell cell (grid, d, i, j)
    return len (cells)
```

```
if __name__ == "__main__":
    grid = []
    for line in sys.stdin:
        line = line.strip()
        grid.append (list (line))
```

~~answer~~ answer = 0

```
for d in range (4):
    for i in range (len (grid)):
        for j in range (len (grid [i])):
            answer = max (answer, launch (grid, d, i, j))
```



Вариант № 2

№6 (продолжение)

is_valid - проверка ~~каждой~~ ячейки $[i][j]$ на входение в границы и отсутствие препятствия.

$next_cell$ - индекс следующей клетки, если робот движется в направлении dx, dy из клетки $[i][j]$

$launch$ - запускать робота из клетки i, j и возвращать из функции количество посещенных клеток

Сложность алгоритма $O(n^2)$ 90 208-?



Вариант № 2

15
import sys
from queue import ~~Priority~~ Priority Queue as PQ
def dek(i, j) ← 05-?
 global home
 global dist
 pq = PQ
 pq.put((0, i, j))
 dist[i][j] = 0
 while not pq.empty():
 d, i, j = pq.get()
 if dist[i][j] < d:
 continue
 for ni, nj in ((i-1, j), (i+1, j),
 (i, j-1), (i, j+1)):
 if ni < 0 or ni >= len(home) or
 nj < 0 or nj >= len(home[ni]):
 continue



Вариант № 2

№5 (программи)

Вывести ~~for~~ for:

new_d = d + 1

if name[ni][nj] == '#' and d == 0

new_d = 0

if dist[ni][nj] == -1 or new_d < dist[ni][nj]:

dist[ni][nj] = new_d

pq.put((new_d, ni, nj))

if __name__ == "__main__":

home = []

dist = []

~~for~~ home.append(list(

dist.append([-1 for

for line in sys.stdin:

home.append(list(line.strip()))

dist.append([-1 for _ in range(len

(len(line.strip()))])

15 (продолжение)

Вариант 2

Шагг работа

```
def len (home):
    n = len(home[0])
    done = False
    for i in range(n):
        for j in range(n):
            if home[i][j] == "#":
                return i, j
```

```
def find (i, j):
    done = True
    break
```

10 не отсюда
или somewhere

почка ~~break~~ done : break

```
result = -1
for i in range(n):
    for j in range(m):
```

```
    if home[i][j] == "#" and dist[i][j] == 0:
```

```
        if (result == -1) or (dist[i][j] < result):
            result = dist[i][j]
```

```
    result = dist[i][j]
```

print result == -1

(изначально 10)

13

Вариант 2

$$\textcircled{I} \quad 12 \geq x^2 \rightarrow 12 \leq (x-1)^2$$

1. Если $|x| \geq 2\sqrt{3}$, то верно всегда

2. Если $|x| \leq 2\sqrt{3}$, то $x \in$ ~~$[-2\sqrt{3}; 2\sqrt{3}]$~~

$$[-2\sqrt{3}; 1-2\sqrt{3}]$$

Общее решение: $x \in (-\infty; 1-2\sqrt{3}] \cup [2\sqrt{3}; +\infty)$

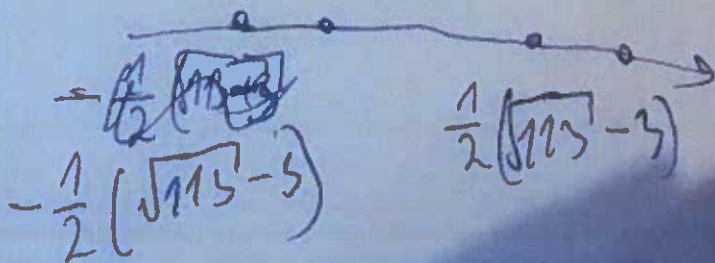
$\textcircled{II}$ Нет целых чисел, если обе стороны
верно

Тогда рассмотрим случаи, когда обе
стороны ложны:

$$\begin{cases} x(x+3) > 26 \\ x^2 + 2x + 1 < 26 \end{cases}$$

$$-1-\sqrt{26}$$

$$\sqrt{26}-1$$



Стороны 11 из
13

Вариант В

Если $\sqrt{3}$ (проучение)

Если там есть целое число, то только ч;

проверим ч и $\frac{1}{2}(\sqrt{115} - 3)$

$11 \sqrt{115}$

$121 > \sqrt{115}$

$4 \sqrt{26} - 1$

$25 < 26$

$\Rightarrow$ ч и округлит только ч, не округлит ч в $\textcircled{1}$

Ответ: ч

18
решение
раскрыто
не полностью,
нет продолжения
фигур. решение

Страница 12 из 13



Вариант 2

✓1

- 1) Все числа $70 < 10$
- 2) Все 4 числа
- 3) Длина C_2 + длина $D_2 = 5$
- 4) $E_2 - F_2 = 0 \Rightarrow E_2 = F_2$
- 5) Одно из E_2, D_2, F_2 - текст из 4)
это не C_2 и не $F_2 \Rightarrow D_2$ - текст
- 6) $F_2 = 5 \Rightarrow E_2 = 5$
- 7) Текст = начало "инкотизер"
- 8) B_2 + длина текста = 4
- 9) сумма чисел 15
- 10) из 3) длина текста 2 "к" все
числа двенадцатилетние по 1) $\Rightarrow$ текст это "ИП"
- 11) из 8) $\Rightarrow B_2 = 2 \Rightarrow$ последние число
это $15 - 5 - 5 - 2 = 3$

Ответ: B - 2 C - 3 D - ИП E - 5 F - 5

- + + +

105

Сумма 13 из
13