



Вариант № 1

Шифр работы

ННЗУ-120

(не заполнять)

Оценка выполнения олимпиадной работы (заполняется проверяющим)										
№ задания	1	2	3	4	5	6	7	8	9	10
Полученный балл	-	3	2	0	20	5	-	-	-	Σ
Подпись проверяющего										
Подпись проверяющего										
Подпись проверяющего										
Подпись проверяющего										
Подпись проверяющего										

№3

$$1) \begin{cases} 52 \leq x^2 \rightarrow 52 \geq (x^2 - 2x + 1) \\ 43 \geq x(x+2) \wedge 43 \leq (x-1)^2 \end{cases}$$

$$2) \begin{cases} 52 \leq x^2 \vee 52 \geq x^2 - 2x + 1 \\ 52 > x^2 \\ x^2 < 52 \\ x \in (-\sqrt{52}; \sqrt{52}) \end{cases}$$

$$3) \begin{cases} 52 > x^2 \\ x^2 < 52 \end{cases} \text{ или } 52 \geq x^2 - 2x + 1 \text{ — равенство имеем: } A \rightarrow B \equiv \bar{A} \vee B$$

$$x^2 - 2x - 51 \leq 0$$

$$D = 4 + 4 \cdot 51 = 4 + 204 = 208$$

$$x_1 = \frac{2 - \sqrt{208}}{2} = \frac{2 - 4\sqrt{13}}{2} = 1 - 2\sqrt{13}$$

$$x_2 = \frac{2 + \sqrt{208}}{2} = \frac{2 + 4\sqrt{13}}{2} = 1 + 2\sqrt{13}$$

$$x \in (-\sqrt{52}; \sqrt{52})$$

$$1 + 2\sqrt{13} > \sqrt{52}$$

$$1 + 4\sqrt{13} + 52 > 52$$

$$1 + 2\sqrt{13} > \sqrt{52}$$

$$x \in (-\sqrt{52}; 1 + 2\sqrt{13})$$

$$x \in (-\sqrt{52}; 1 + 2\sqrt{13})$$

$$x \in (-\sqrt{52}; 1 + 2\sqrt{13})$$

$$x \in (-\sqrt{52}; 1 + 2\sqrt{13})$$

$$x \in (-\sqrt{52}; 1 + 2\sqrt{13})$$

$$x \in (-\sqrt{52}; 1 + 2\sqrt{13})$$

$$x \in (-\sqrt{52}; 1 + 2\sqrt{13})$$



Вариант № 1

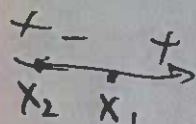
1) $43 \geq x^2 + 2x$

$x^2 + 2x - 43 \leq 0$

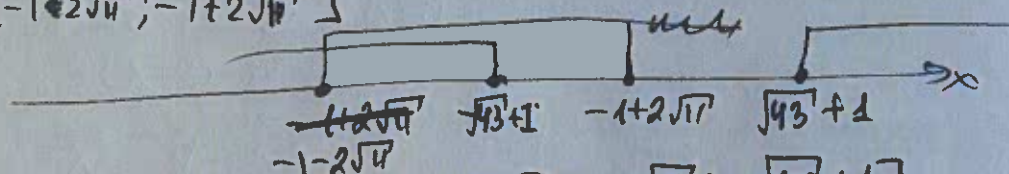
$D = 4 + 4 \cdot 43 = 176$

$x_1 = \frac{-2 + \sqrt{176}}{2} = -1 + 2\sqrt{11}$

$x_2 = \frac{-2 - \sqrt{176}}{2} = -1 - 2\sqrt{11}$

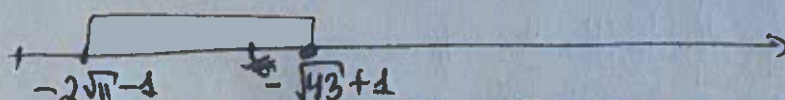
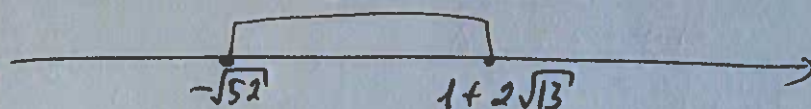


$x \in [-1 - 2\sqrt{11}; -1 + 2\sqrt{11}]$



$x \in [-1 - 2\sqrt{11}; -\sqrt{43} + 1]$

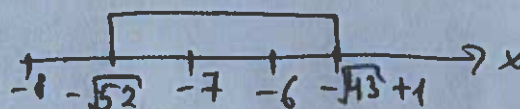
Найдем пересечение решений:



в итоге: $x \in [-6; -\sqrt{43} + 1]$

$-6 < -\sqrt{43} + 1 < -5$

$-\sqrt{43} + 1 > -6, \text{ а } -6 < -\sqrt{52} < -7$



Ответ: -6

ИЗ

Возьмем, например, второе слово: «ТЕБЯ», оно превращается в «ПВ.6Э» и «Уё'В).». Можно заметить, что в 1-м случае преобразование на 3 знака ($E \xrightarrow{-3} B$), а во 2-м случае — на 1 вперед ($E \xrightarrow{+1} \dot{E}$). т.к. в 1-м случае «Т» перешла в «Э», а «Б» перешла в «.», то аналогично получится так: «. ? АБВ... ЭЮЯ»



Вариант № 1

Требуется также составить таблицу. Прок по условию -
оригинальный алфавит слова - переводят при введении 1,
а справа - при -1.

1	Оригинал	-1
10	?	?
2	?	A
1	A	B
?	B	D
A	B	C
...	C	A
7	?	10
4	10	2
6	я	?
?	?	?

Найдем правило шифрования. По условию это
многозначен 4-5 степеней: $f(x) = kx + b$. Зная что

$$\begin{cases} f(1) = -3 \text{ и } f(-1) = 1, \text{ получим:} \\ \begin{cases} 1 - k + b = -3 \\ -3 = k + b \end{cases} \Rightarrow \begin{cases} b = -3 - k \\ 1 = -k - 3 - k \end{cases} \Rightarrow \begin{cases} b = -3 - k \\ 2k = -4 \end{cases} \Rightarrow \begin{cases} k = -2 \\ b = -1 \end{cases} \end{cases}$$

$f(x) = -2x - 1$, - переводит слева по алфавиту,
где X - введенное число

Ответ: $f(x) = -2x - 1$ - переводит слева по алфавиту,
Алфавит указан выше.

№6.

Воспользуемся функцией шифрования,
в массиве $d[i; j]$ будем хранить окончательное
значение при переходе по массиву $d[i; j]$ страница 2 из 7

оригинал
по алфавиту
а по условию

50

15

что такое
в поиме

25

Вариант № 1

Также выполняемое различное функционирование.
Нашими профессорами.

```

n = int(input())
m = int(input())
or = [] # yes xamam uacub
for i in range(0, n): # blog garame
    for j in range(0, m):
        or.append(i, j)
        or.append(i, j)
        or.append(i, j)
        S = input().split()
        for j in range(len(S)): # True xamam uacub
            or[i][j] = True if S[j] == '~' else False
        max_v = 0 # yes xamam uacub
        for i in range(n):
            for j in range(m):
                if or[i][j] == False:
                    continue # kunge xamam uacub
                d[i][j] = get_d(i, j, [0, 0])
                if d[i][j] > max_v:
                    max_v = d[i][j]
print(max_v)

```

```

def get_d(i, j) return 0
def get_d(i, j, dir):
    if d[i][j] != -1:
        return d[i][j]
    else:
        return max(
            get_d(i, j, 0),
            get_d(i, j, 1),
            get_d(i, j, 2),
            get_d(i, j, 3)
        )
    d[i][j] = get_d(i, j, 0) + 1
    return d[i][j]

```




Вариант № 1

else: # если вапери ~~того~~, ~~поворачиваем~~ направо

dir_right = [0 if dir[0] != 0 else dir[1]] [0 if dir[1] != 0 else -dir[0]]

if ar[i + dir_right[0]][j + dir_right[1]] == True: # ~~правим~~ ^{свободно}

return get_d(i + dir_right[0], j + dir_right[1], dir_right) ^{мы слева}
~~затягиваем~~ ^{затягиваем} ~~от~~ ^{от} ~~клетки~~ ^{клетки} ~~и~~ ^и ~~справа~~ ^{справа}

Рассмотрим, как ~~направился~~ ^{направление} направо.
Если первоначальное направление было $[x_0; y_0]$,

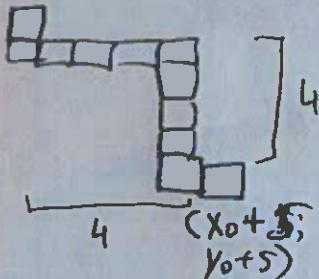
то после поворота направо получим: $\begin{bmatrix} x_0 \neq 0 \Rightarrow 0, y_0 \neq 0 \Rightarrow 0 \\ x_0 = 0 \Rightarrow y_0, y_0 = 0 \Rightarrow -x_0 \end{bmatrix}$



Например, $[0; 1] \xrightarrow{\text{вверх}} [1; 0] \xrightarrow{\text{вправо}} [0; -1] \xrightarrow{\text{вниз}}$

Например такой:

$(x_0; y_0)$



$r = 5 + 5 - 1 = 9$

Напишем программу:

~~x=30~~
~~y=15~~

$x=30$

$y=15$

размер карты

for i in range(x):

for j in range(y):

ar[i][j] = False if sc[j] == '#' else True

v.append([i, j]) # добавим координаты

if sc[j] == '#': # если стена

v.append([i, j])

ar = [[False for i in range(x)] for j in range(y)]
v = [] - массив точек пути



Вариант № 1

```
v1 = dfs(v[0]) # найти мин-д остров замкнутом dfs
# любой точн. суши
v2 = [0] # any coord for coord in v if coord not in v1 # там же
# остров два земли все точки
# земли которые не принадлежат
# острову!
min_d = 1000000000000
for c1 in v1: # перебираем координаты остров 1
    for c2 in v2: # перебираем координаты острова 2
        d = abs(c1[0]-c2[0]) + abs(c1[1]-c2[1]) - 1 # расстояние
        if d < min_d:
            min_d = d
```

print(min_d)

Функция dfs(c) - находит номер в глубине от координат. Ее можно реализовать так:

```
def dfs(coord):
    x = coord[0]
    y = coord[1]
    if (x, y) in visited:
        return visited
    visited.append(coord) # добавляем координату в список посещенных
    if or[x+1][y] == True: # если суша рядом
        dfs(x+1, y, visited) # делаем dfs от точки суша
    if or[x-1][y] == True: # если суша рядом
        dfs(x-1, y, visited)
    if or[x][y+1] == True: # если суша сверху
        dfs(x, y+1, visited)
    if or[x][y-1] == True: # если суша снизу
        dfs(x, y-1, visited)
```

return visited # возвращаем список посещенных точек (или все - таеть острова)

У-за по блоку кода
список координат острова O(x,y), где x,y - координаты
острова (x=30; y=15)

Страница 6 из 7

Наименование сети	Число хостов	Маска подсети	Ресурсы	Шлюз по умолчанию	Диагностика
П.к. корп.	1000	/18		128.0.0.0	
Ш.к. корп.	500	/18		128.3.14.8	
С.к. корп.	250	/18		128.4.110.	
Будущие	30	/18		128.5.1.4.	
Ресурсы	14	/18		128.5.2.2.	
IT-отдел	4	/18		128.5.2.6	

$$1000_{10} = \underbrace{1111111000}_3 \underbrace{14}_4 \underbrace{8}_8$$

$$1050_{10} = \underbrace{100000}_4 \underbrace{11010}_1 \underbrace{10}_{10}$$

$$1300_{10} = \underbrace{10100010100}_5 \underbrace{1}_1 \underbrace{4}_4$$

$$1314_{10} = \underbrace{10100100010}_5 \underbrace{2}_2 \underbrace{2}_2$$

$$1318_{10} = \underbrace{10100100110}_5 \underbrace{2}_2 \underbrace{6}_6$$

128.0.0.0
128.3.14.8

