



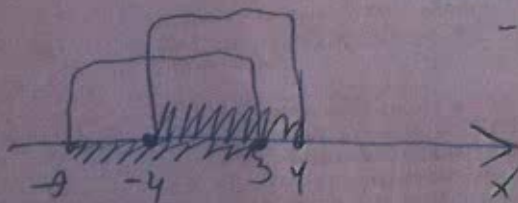
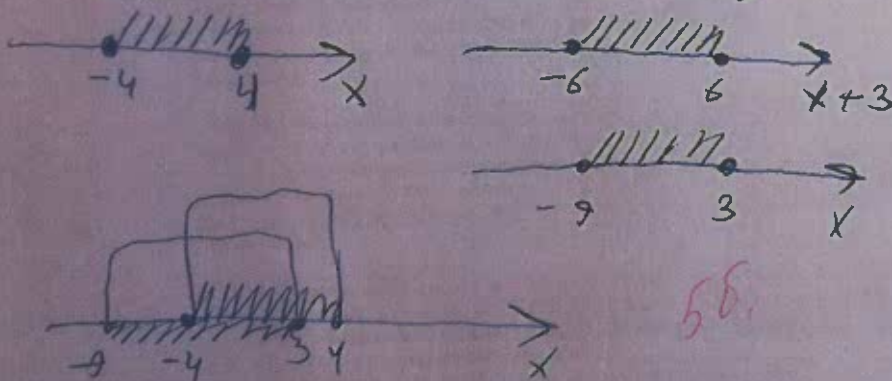
Вариант № 4

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Полученный балл	10	5	5	—	20	35					345
I проверка	Подпись проверяющего			Σ баллов прописью	тридцать три						
I проверка	Подпись проверяющего			Σ баллов прописью	тридцать три						
II проверка	Подпись проверяющего			Σ баллов прописью	сорок пять						
II проверка	Подпись проверяющего			Σ баллов прописью	сорок пять						

№3 { $48 \leq 2x^2 \rightarrow 48 \geq (x^2 + 6x + 9)$
 $57 \leq x(3x-5) \equiv 57 \geq (x-2)^2$

① $(48 \leq 2x^2) + (48 \geq (x+3)^2) = 1$
 $(24 \geq x^2) + (48 \geq (x+3)^2) = 1$



$x \in [-9; 4]$

② $(3x^2 - 5x - 57)(-x^2 + 4x + 53) \geq 0$
 $\begin{cases} 3x^2 - 5x - 57 \leq 0 \\ -x^2 + 4x + 53 \leq 0 \end{cases} \quad \begin{cases} 3x^2 - 5x - 57 \geq 0 \\ -x^2 + 4x + 53 \geq 0 \end{cases}$
 $x \in [-3; 5] \quad x \in (-\infty; -5] \cup [9; \infty)$

$(3x^2 - 5x - 57)(x^2 - 4x - 53) \leq 0$
 $\begin{cases} 3x^2 - 5x - 57 \leq 0 \\ x^2 - 4x - 53 \geq 0 \end{cases} \quad \begin{cases} 3x^2 - 5x - 57 \geq 0 \\ x^2 - 4x - 53 \leq 0 \end{cases}$
 $\Rightarrow x \in [-3; 5] \quad \Rightarrow x \in [-5; 9]$

$x \in [-5; -4]$

$\Rightarrow$ max case $x = -4$

Оптимально $x = -4$

$x \in [-5; -4]$

$\cup [-6; 3]$



Вариант № 4

[Pascal ABC.net]

```
n5 program ostrova;
var i, j, ig, jg, sum1, sum2, rmin: int64;
a: array [0..16, 0..31] of integer;
s: string;
f: int64;
procedure ostrov1(x, y: integer);
begin
  if a[x, y] = 1 then
  begin
    if a[x-1, y] = 2 then
    begin
      a[x-1, y] = 1;
      ostrov1(x-1, y);
    end;
    if a[x+1, y] = 2 then
    begin
      a[x+1, y] = 1;
      ostrov1(x+1, y);
    end;
    if a[x, y-1] = 2 then
    begin
      a[x, y-1] = 1;
      ostrov1(x, y-1);
    end;
    if a[x, y+1] = 2 then
    begin
      a[x, y+1] = 1;
      ostrov1(x, y+1);
    end;
  end;
end;
```

↓ продолжение



Вариант № 4

~5 (программирование)

begin

Гmin:=50;

for i:=1 to 15 do

begin

readln(s);

for j:=1 to 30 do

begin

if (s[j] = '~') then a[i,j]:=0

else

if (s[j] = '#') and (f=0) then

begin

a[i,j]:=1;

sum1:=i;

sum2:=j

f:=1;

end

else

a[i,j]:=2;

end;

end;

ostrou1(sum1, sum2);

loop 5 do

Гmin:=50;

for i:=1 to 15 do

for j:=1 to 30 do

for ir:=1 to 15 do

for jr:=1 to 30 do

begin

if (a[i,j]=1) and (a[ir,jr]=2) then

Гmin:=min(Гmin, max(i,ir)+max(j,jr)-min(i,ir)-
- (min(j,jr)-1);

~~10.5~~ (20)

За сумму
постав

по 2.5



Вариант № 4

~5 (money)
println("min");
end.

~2

N°	°	,	?	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
1	°	ю	я	°	?	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
-1	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ

N°	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
	Ъ	Ы	Ь	Э	Ю	Я	°	,	?	А	Б

при 1 сдвиг на 5 вправо
при -1 сдвиг на 3 влево

58.

Ю- шифр $\Rightarrow -4a+1$
вводится





Вариант № 4

Исследова значения формул, мы можем сделать вывод,
что узор 1 и узор 3 и их сумма (0; 10)
Проведя анализ, в текстовых формулах заметим,
что ~~ИИ34-36~~ $D_2 = 2 \cdot 10$.
Приведем оставшиеся формулы полными итер:



- 1) (0; 10)
- 2) 4 итер и 1 текстовая
- 3) G_2 и D_2 значений 8 2
- 4) E_1 и F_2 - итер
- 5) D_2 - итер
- 6) $F_2 - 1 = E_2$
- 7) гипотеза $D_2 = 2 \cdot 10$
- 8) $B_2 = 1$
- 9) $sum = 10 \Rightarrow C_2 = 7$

1	7	ер	1	1
---	---	----	---	---



[python 3.10]

Вариант № 4

```
№6 def checkVis (x, y, k, d, mp):  
    return 0 <= x < k and 0 <= y < m and (mp[x][y] == 1)  
  
    trajectories = [[1, 0], [0, 1], [-1, 0], [0, -1]]  
  
    def square (k, d, mp, sx, sy, tr):  
        v = [[False for k in range(k)] for j in range(m)]  
        for min range(s):  
            sum = 0  
            while not (v[cx][cy][cb]):  
                v[cx][cy][cb] = True  
                tx = cx + trajectories[cb][0]  
                ty = cy + trajectories[cb][1]  
                if checkVis(tx, ty, k, d, mp):  
                    cy = ty  
                    cx = tx  
                else:  
                    cb = (cb + 1) % 4  
  
            for i range(k):  
                for j range(d):  
                    if (v[k][d][0] or v[k][d][1] or  
                        v[k][d][2] or v[k][d][3]):  
                        sum += 1
```



Вариант № 4

```
def findsquare (k, d, mp):  
    b = -1  
    for i in range(k+1-1):  
        for j in range(d):  
            for k in range(4):  
                if (mp[i][j] == '#'):  
                    continue  
                S = findsquare square (k, d, mp, i, j, k)  
                b = max(b, S)  
    return b
```

```
n, m = map(int, input().split())  
cordort = [[0 for i in range(m)]  
for x in range(m):  
    str = list(input())  
    for y in range(n):  
        cordort[y][x] = str[y]  
print(findsquare(n, m, cordort))  
for x in range(m):  
    cordort[1][x] = cordort[1][x]
```

5.
частичное
решение
ошибка не
ошибка.