



Олимпиада школьников «Гранит науки»
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Шифр работы ШЗЧ-155
(не заполнять)

Вариант № 1

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

№ задания		1	2	3	4	5	6	7	8	9	10	Σ
Полученный балл		—	3	10	5	25	0	—	—	—	—	43
I проверка	Подпись проверяющего				Σ баллов прописью	сорок три						
	Подпись проверяющего				Σ баллов прописью	сорок три						
II проверка	Подпись проверяющего				Σ баллов прописью	сорок три						
	Подпись проверяющего				Σ баллов прописью							



Вариант № 1

```
#6 #include <iostream>
#include <vector>
#include <string>
#include <algorithm>

int n, m, count0;
bool used[100][100];
int a[100][100];
vector<int> ans;

void dfs(int x, int y, int &count0) {
    used[x][y] = true;

    count0++;
    if (y != m-1 && a[x][y+1] && !used[x][y+1]) {
        dfs(x, y+1); dfs(x, y+1, count0);
    }
    else if (x != 0 and && a[x-1][y] && !used[x-1][y]) {
        dfs(x-1, y, count0); dfs(x-1, y, count0);
    }
    else if (y != 0 && a[x][y-1] && !used[x][y-1]) {
        dfs(x, y-1, count0); dfs(x, y-1, count0);
    }
    else if (x != n-1 && a[x+1][y] && !used[x+1][y]) {
        dfs(x+1, y, count0);
    }
}

int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            char ch;
            cin >> ch;
            if (ch == '~') {
                a[i][j] = 1;
            }
        }
    }
}
```




Вариант № 1

№6 (продолжение)

```
for (int i=0; i<n; i++) {
    for (int j=0; j<m; j++) {
        if (!used[i][j] && a[i][j] == 1) {
            int count = 0;
            dfs(i, j);
            ans.push_back(count);
        }
    }
}
sort(ans.begin(), ans.end());
cout << ans[0];
}
```



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

Так как $\wedge$ обозначает and, то (2) можно разбить на систему.

$$\begin{cases} 52 > x^2 \text{ or } 52 \geq (x^2 + 1 - 2x) \\ 43 \geq x(x+2) \\ 43 \leq (x-1)^2 \end{cases} \quad \text{or-объединение } \Rightarrow$$

$$\Rightarrow \begin{cases} \begin{cases} x^2 - 52 > x^2 \\ 52 \geq x^2 + 1 - 2x \\ 43 \geq x(x+2) \\ 43 \leq (x-1)^2 \end{cases} & \begin{cases} x^2 - 52 < 0 \\ x^2 - 2x - 51 \leq 0 \quad (1) \\ x^2 + 2x - 43 \leq 0 \quad (2) \\ x^2 - 2x - 42 \geq 0 \quad (3) \end{cases} \end{cases}$$

Разложим на множители

$$(1) x^2 - 2x - 51 = 0$$

$$D = 4 + 204 = 208$$

$$x_{1,2} = \frac{2 \pm \sqrt{208}}{2} = 1 \pm \sqrt{13}$$

$$x_1 \approx -6,2$$

$$x_2 \approx 8,2$$

$$(2) x^2 + 2x - 43 = 0$$

$$D = 4 + 172 = 176$$

$$x_{1,2} = \frac{-2 \pm \sqrt{176}}{2} = -1 \pm 2\sqrt{11}$$

$$x_1 \approx -7,6$$

$$x_2 \approx 5,6$$

Вариант № 1

№3 (продолжение)

(3) $x^2 - 2x - 42 = 0$

$D = 4 + 168 = 172$

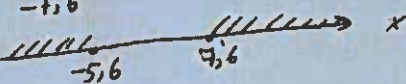
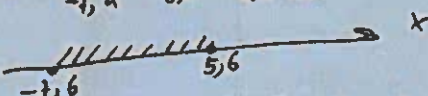
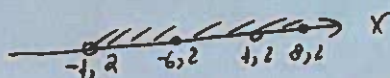
$x_{1,2} = \frac{2 \pm \sqrt{172}}{2} = 1 \pm \sqrt{43}$

$x_1 \approx -5,6$

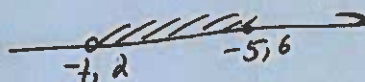
$x_2 \approx 7,2$

$\sqrt{52} \approx 7,2$

$$\begin{cases} (x - \sqrt{52})(x + \sqrt{52}) < 0 \\ (x + 6,2)(x - 8,2) \leq 0 \\ (x + 7,6)(x - 5,6) \leq 0 \\ (x - 7,6)(x + 5,6) \geq 0 \end{cases}$$



Общие решения



Наибольшее целое $x = -6$

Ответ: -6

Для шифра $n=1$

1

$n=1$

Введенные символы	Возвращенные символы
А	К
Б	Ы
В	Г
Г	В
Д	З
Е	?
Ё	И
Ж	М
З	Ц
И	Е
Й	Р
К	Э
Л	Ж
М	О
Н	Н
О	О
П	П
Р	Н

Введенные символы	Возвращенные символы
С	Р
Т	Т
У	У
Ф	Ц
Х	Х
Ц	Ц
Ч	Ч
Ш	Ш
Щ	Щ
Ъ	Я
Ы	Ы
Ь	Ь
Э	Э
Ю	Ю
Я	Я



Вариант № 1

Для шифра $n = -1$

у	х	о
ч	ш	ь
ш	ж	ю
щ	ъ	х
ъ	ы	ё
ы	в	б
ь	э	п
э	ю	з
ю	я	г
я	ц	к
ц	м	
к	а	
а	е	

№2 (продолжение)

Для шифра $n = 1$

Введен. шифр	Возврат. шифр
?	?
?	?
?	?

Для шифра $n = -1$

Введен. шифр

Возврат. шифр

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	А
В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Б	
Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Г		
Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Д			
Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Е				
Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ж					
З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	З						
И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	И							
Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Й								
К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	К									
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Л										
М	Н	О	П	Р	С	Т	У	Ф	Х	М											
Н	О	П	Р	С	Т	У	Ф	Х	Н												
О	П	Р	С	Т	У	Ф	Х	О													
П	Р	С	Т	У	Ф	Х	П														
Р	С	Т	У	Ф	Х	Р															
С	Т	У	Ф	Х	С																
Т	У	Ф	Х	Т																	
У	Ф	Х	У																		
Ф	Х	Ф																			
Х	Х																				

Известно, что шифрование происходит по принципу, состоящего из одного в первой степени и одного свободного знака. Таким образом, каждый символ исходного текста заменяется на символ шифрованного текста с соотв. принципом.

Для шифра 1 мы можем записать принцип в виде $Bx + 1$

Для шифра -1 мы можем записать принцип в виде $-Bx + 1$

Тогда символы исходного текста будут заменены на символы с соотв. принципом

правильно
35

Вариант № 1

Название подсети	Число узлов в сети	Адрес подсети	Маска - префикс	Дефигирированная маска	Шлюз по умолчанию	Адреса в сети по умолчанию
Главный корпус	1000	172.16.0.0	122	255.255.252.0	172.16.0.1	172.16.0.1 - 172.16.3.254
Инженерный корпус	500	172.16.4.0	123	255.255.254.0	172.16.4.1	172.16.4.1 - 172.16.5.254
Учебный корпус №2	250	172.16.6.0	124	255.255.255.0	172.16.6.1	172.16.6.1 - 172.16.6.254
Библиотека	30	172.16.6.32	127	255.255.255.224	172.16.6.33	172.16.6.33 - 172.16.6.62
Посты охраны	14	172.16.6.48	128	255.255.255.240	172.16.6.49	172.16.6.49 - 172.16.6.62
IT-отдел	4	172.16.6.56	129	255.255.255.248	172.16.6.57	172.16.6.57 - 172.16.6.62

(5)

55-?
сеть правильная, но
сеть отключена



Вариант № 1

№5. Используйте алгоритм поиска в ширину.
Будем начинать с 1 острова и двигаться по соседним клеткам, пока не
дойдем до второго острова.

← def find_minimum(gr):

start_row, start_col = None, None

for row in range(len(gr)):

for col in range(~~len(gr)~~) (len(gr[0])):

if gr[row][col] == '#':

start_row, start_col = row, col
break

if start_row is not None:
break

quene = [(start_row, start_col)]

visited = set(quene)

while quene:

row, col = quene.pop(0)

for d_row, d_col in [(1, 0), (-1, 0), (0, 1), (0, -1)]:

new_row = d_row + row

new_col = d_col + col

if (new_row, new_col) in visited or not is_valid_cell(gr, new_row, new_col):

continue

visited.add((new_row, new_col))

if gr[new_row][new_col] == "#":

quene.append((new_row, new_col))

end_row, end_col = None, None

for row in range(len(gr)):

for col in range(len(gr[0])):

if gr[row][col] == '#' and (row, col) not visited:

end_row, end_col = row, col

break

if end_row, end_col = row, col

break

25



Вариант № 1

```
quence = [(end-row, end-col)]
visited = set(quence)
while quence:
    row, col = quence.pop(0)
    for d-row, d-col in [(1,0), (-1,0), (0,1), (0,-1)]:
        new-row = d-row + row
        new-col = d-col + col
        if (new-row, new-col) in visited or not is_valid_cell(gr, new-
            row, new-col):
            continue
        visited.add((new-row, new-col))
        if gr[new-row][new-col] == "#":
            quence.append((new-row, new-col))

count = 0
for row in range(len(gr)):
    for col in range(len(gr[0])):
        if gr[row][col] == "." and (row, col) in visited:
            count += 1
    return count

gr = []
for i in range(15):
    row = input().strip()
    gr.append(row)

minimum_cell = find_minimum_cell(gr)
print(minimum_cell)
```