Олимпиада школьников «Гранит науки» Шифр работы Ч434-1Ч5
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

Вариант № 3

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Полученный балл	–	–	7	10	~~25~~ 20	~~20~~ 10					~~47~~ 62

I проверка	Подпись проверяющего		Σ баллов прописью	сорок семь
	Подпись проверяющего		Σ баллов прописью	сорок семь
II проверка	Подпись проверяющего		Σ баллов прописью	шестьдесят два.
	Подпись проверяющего		Σ баллов прописью	

4. Название подсети | Число хостов в сети | Адрес подсети | Маска-префикс | Дес. маска | Шлюз по умолч. | Диап. дост. адр. для узлов

Название подсети	Число хостов в сети	Адрес подсети	Маска-префикс	Дес. маска	Шлюз по умолч.	Диап. дост. адр. для узлов
Производственный корпус	1000	191.255.0.0 ±	/22 +	255.255.252.0 +	191.255.0.1 ±	191.255.0.1 – 191.255.3.254 ±
Научно-исследовательский корпус	500	191.255.4.0 ±	/23 +	255.255.254.0 +	191.255.4.1 ±	191.255.4.1 – 191.255.5.254 ±
Лабораторный корпус	250	191.255.6.0 ±	/24 +	255.255.255.0 +	191.255.6.1 ±	191.255.6.1 – 191.255.6.254 ±
Библиотека	30	191.255.7.0 ±	/26 +	255.255.255.192 +	191.255.7.1 ±	191.255.7.1 – 191.255.7.62 ±
Администрация	14	191.255.7.64 ±	/27 +	255.255.255.224 +	191.255.7.65 ±	191.255.7.65 – 191.255.7.94 ±
IT-отдел	4	191.255.7.96 ±	/29 +	255.255.255.248 +	191.255.7.97 ±	191.255.7.97 – 191.255.7.102 ±

Класс B – это адреса в диапазоне от 128.0.0.0 до ~~191~~ 205, 191.255.255.255, где сеть задаётся первыми 16 битами (то есть первыми двумя числами), соответственно, второй сетью будет 128.1.0.0.

(не заполнять)

10	Σ

③ $\begin{cases} 34 \le x^2 \to 34 \ge (x^2+8x+16) \\ 42 \le x(2x+7) \cap 42 \ge (x-3)^2 \end{cases}$

1) Выражение ~~верно~~ $34 \le x^2 \to 34 \ge (x^2+8x+16)$ верно, при:

- $|x| < \sqrt{34}$, если обе части возведём в степень $\frac{1}{2}$.
- если $|x| \ge \sqrt{34}$, то $x \in [-4-\sqrt{34}; -\sqrt{34}]$; $\sqrt{34} \sim 5{,}8$

2) Проверим способом подстановки, какое из решений в пункте 1 будет верным:

возьмём ~~наибольшее, тогда~~ ближайшее к $5{,}8 \sim 5$.
и
$5{,}8 \sim 6$

при $x=5$: $42 \le 5(5\cdot 2+7) \cap 42 \ge (5-3)^2$

$42 \le 17\cdot 5 \cap 42 \ge 4$ – верно.

$42 \le 85 \cap 42 \ge 4$.

$34 \le 5^2 \to 34 \ge 5^2+8\cdot 5+16$ – верно.

7 б. не график

при $x=6$: система неверна, следовательно, наибольшее целое число – 5.

Ответ: 5.

(3) (6) Так как робот движется по матрице размером N×M - вперёд пока нет препятствия, и вправо поворачивает вправо, если перед ним препятствие, то совершенно логично 1) использовать алгоритм динамического программирования, который будет искать наибольший путь робота, то есть площадь.

Код на языке C++ для большей оптимизации выполнения работы

на следующей странице.

1) Создадим структуру Robot.
2) в next мы делаем шаг, если нельзя сделать, возвращаем false.
3) в clean запускаем робота и считаем очищённые клетки, возвращаем количество клеток, которые посетил робот хотя бы 1 раз
4) В main в циклах проходимся по полю и запускаем робота из каждой свободной клетки.
5) В answer храним ответ. Берём максимум из пройденных и очищенных клеток. То есть, как гласит алгоритм ДП - ищем максимум возможных вариантов (в нашем случае клеток).

Олимпиада школьников «Гранит науки» Шифр работы 11134-146
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

Вариант № 3

```
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <set>
#include <tuple>

using Home = std::vector<~~string~~ std::string>;

struct Robot {
    static constexpr int di[] = {-1,0,1,0};
    static constexpr int dj[] = {0,1,0,-1};
    int direction, i, j;
    Home *home;
    bool next() {
        for(int newD=direction; newD<direction+4; newD++) {
            int newI = i + di[newD%4];
            int newJ = j + dj[newD%4];
            if (0 <= newI && 0 <= newJ && newI < home->size() &&
                newJ < (*home)[i].size() && (*home)[newI][newJ] == '~') {
                i = newI;
                j = newJ;
                direction = newD%4;
                return true;
            }
        }
    }
```

Вариант № 3

```
int clean() {
    Robot backup = *this;
    std::set<std::tuple<int,int,int>> orientations;
    std::set<std::pair<int,int>>cleanedCells;
    while (true) {
        auto orientation$ = std::make_tuple(i,j,direction);
        if(orientations.count(orientation)) {
            break;
        }
        orientations.insert(orientation);
        cleanedCells.insert(std::make_pair(i,j));
        if(!next()) break;
    }
    *this = next();
    *this = backup;
    return cleanedCells.size();
}
};
```

Олимпиада школьников «Гранит науки» Шифр работы Ш34-146
(не заполнять)
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Вариант № 3

(3)
```
int main() {
    ~~Ком~~
    cin.tie(0);
    cout.tie(0);
    ios::sync_with_stdio(0);

1)  Home grid;

    std::string line;
    while (std::cin>>line) {
        grid.push(line);
    }

    Robot robot;
    robot.home = &grid;
    ~~int answer%=0;~~
    int answer=0;
    for(robot.direction=0; robot.direction<4; robot.direction++){
        for(robot.i=0; robot.i<grid.size(); robot.i++){
            for(robot.j=0; robot.j<grid[robot.i].size(); robot.j++){
                answer = std::max(answer, robot.clean());
            }
        }
    }
    std::cout<<answer;
}
```

for.

(20) частичное решение.

```
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <set>
#include <tuple>
#include <queue>

using Country = std::vector<std::string>;
using Dist = std::vector<std::vector<int>>;

struct Crawler() {
    static constexpr int di[] = {-1,0,1,0};
    static constexpr int dj[] = {0,1,0,-1};

    Country *home;
    void dijkstra(int i, int j, Dist &dist) {
        std::queue<std::pair<int,int>> q, qBackup;
        dist[i][j]=0;
        q.push(std::pair{i,j});
        qBackup(std::pair{i,j});
        while (q.size()>0) {
            auto [i,j] = q.front();
            q.pop();
            for(int d=0; d<4; d++) {
            int newI = i+di[d];
            int newJ = j+dj[d];
```

Вариант № 3

```
            if( 0<=new[1] && 0<=new[0] && new[1]<home->size()
(5) && new[0]) < (*home)[new[1]].size()) {
(3)                 if (dist[new[1]][new[0]] != -1 && dist[new[1][new[0]]
    <=curDist[ ] +1) {
1)                      continue;
                    }
                    dist[new[1]][new[0]] = curDist+1;
                    pq.push(std::make_tuple(-dist[new[1]][new[0]], new[1],
2                                                   new[0]));
                }
            }
        }
    }
};

int main(){
    cin.tie(0);
    cout.tie(0);
    ios::sync_with_stdio(0);
    Dist dist;
    Country grid;
```


}

```
std::string line;
while(std::cin>>line) {
    grid.push_back(line);
    dist.emplace_back(line.size(),-1);
}
Crawler c;
c.home = &grid;
bool finished=false;
for(int i=0; grid.size()>i && !finished; i++) {
    for(int j=0; grid[i].size<j && !finished; j++) {
        if(grid[i][j] == '#'){
            c.dijstra(i,j,dist);
            finished=true;
        }
    }
}
int answer=-1;
for(int i=0; i<grid.size(); i++){
    for(int j=0; j<grid[i].size(); j++) {
        if(dist[i][j] == -1 && grid[i].strand
```

Вариант № 3

```
//
    dist[i][j] == 0 || dist[i][j] != '#') {
    continue;
}
if (answer == -1 || dist[i][j] < answer) {
    answer = dist[i][j] - 1;
}
}
}
std::cout << answer;
}
```

доб.

Мы используем алгоритм дейкстра – нахождение минимальных (расстояний между клетками в графе.

Другими словами: кратчайший путь в графе.

1) Создадим как бб задаче структуру и установим dist[x][y] для каждой клетки в том же острове, что и клетка
2) Далее найдём dist[x][y] для ~~расстояние от~~ ближайшего расстояние от одного острова до другого
3) В while(q.size()) алгоритм bfs – поиск в глубину в графе.
4) В main считаем расстояние (кол-во ходов). Кол-во слов, которые нужно заменить.

12

Вариант № 3

① Так как формулы сместились на 1 клетку вправо и 1 клетку вниз, то:

1) В диапазоне B2:F2, чисел $\left[\begin{array}{l} x \le 0 \\ x \ge 10 \end{array}\right.$ нет, тогда:

2) среди B2:F2 только 4 числа. Сумма этих четырёх чисел равна 16.

3) • Если C2 - текст, то (длина C2) + (длина числа C2)
+
• Если D2 - текст, то (длина D2) + (длина числа D2)
=
③ сумма = 3.

~~то есть длина~~.

4) E2 - F2 = 0 => E2 = F2 (по свойству). F2 и E2 не текст, тогда:
D2 - текст, потому что в ячейке в формуле в ошибка.

5) C2 не является текстом тогда и только тогда, когда ЕСЛИ(ЕТЕКСТ) выдаст пустую строку, следовательно в предпоследней формуле (длина B2 + D2) = 9, при этом не D2 ≤ B2.

Вариант № 3

8) 11 формула гласит, что D2 – префикс "+ из". => D2 это ТИ, то есть все числа из 1 знака.

5) Далее E2 = F2 = 2 (так как это вторая позиция)

10) Ответом будет значения в 5 ячейках:

1) данка D2+B2=5+2=7
2) B2=7
3) C2=5
4) D = ТИ (8 пункт)
5) E2 = 2
6) F2 = 2

	B	C	D	E	F
2	7	5	ТИ	2	2

10б.

Ответ: 75ТИ22.