



ВАРИАНТ № 7

Оценка выполнения олимпиадной работы (заполняется проверяющим)												
Первая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	—	5	—	20	23	21					69
	Σ баллов прописью	шестьдесят девять (69)						Подпись проверяющего		Иван		45
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	—	5	—	16	23	21					65
	Σ баллов прописью	шестьдесят пять (65)						Подпись проверяющего		С.В.		
								Подпись проверяющего				

Задача 6

Бинарным поиском будем подбирать длину стороны робота и проверять*: сможем ли робот со стороной m пройти трассу.

*Проверка: (bfs): сможем ли робот ~~перейти~~ переместиться в какое-либо направление. Если сможем — рекурсивно переходим. Если каким-либо путём сможем дойти до правой стороны поля, функция ~~def~~ bfs вернёт true. Также был введён массив used, который позволяет избежать от «зацикливания» программы и прохождения одного и того же маршрута несколько раз. ~~vector<vector<used bool>> used хранит было~~
~~ли использовано поле размером ans x ans, которое~~
~~хранит~~ ~~и~~ ~~used[i][j]~~ отвечает за: «Был ли использовано поле размером $k \times k$, с левыми нижними



Вариант № 7

уном в клетке i, j ".
код на языке C++:

```
#include <iostream>
#include <vector>

using namespace std;

int n, m;
vector<vector<char>> v; // name
bool check(vector<vector<bool>> &used, int &i, int &j){
    if (used[i][j]) return false;
    bool flag = true;
    for (int x = i; x < i + k; ++x)
        for (int y = j; y < j + k; ++y){
            if (v[x][y] == '#') flag = false;
        }
    return flag;
}

bool bfs (int i, int j, int k, vector<vector<bool>> used){
    if (i + k > m) return true;
    bool res = false;
    if (check(i + 1, j)) {
        used[i + 1][j] = true;
        if (bfs(i + 1, j, k, used)) return true;
        used[i + 1][j] = false;
    }
}
```



Вариант № 7

```
if (check(used, i-1, j)) {  
    used[i-1][j] = true;  
    if (bfs(j-1, i, k, used)) if (bfs(i-1, j, k, used)) return true;  
    used[i-1][j] = false;  
}  
if (check(used, i, j+1)) {  
    used[i][j+1] = true;  
    if (bfs(i, j+1, k, used)) return true;  
    used[i][j+1] = false;  
}  
if (check(used, i, j-1)) {  
    used[i][j-1] = true;  
    if (bfs(i, j+1, k, used)) if (bfs(i, j-1, k, used)) return true;  
    used[i][j-1] = false;  
}  
return false;  
}  
  
int main() {  
    cin >> n >> m;  
    v.resize(n+1);  
    for (int i = 1; i <= n; ++i) {  
        v[i].push_back('#');  
        for (int j = 1; j <= m; ++j) {  
            char cur;  
            cin >> cur;  
            v[i].push_back(cur);  
        }  
    }  
}
```



Вариант № 7

```
    }  
    }  
    for(int i=0; i<=m; ++i){  
        v[0].push_back(1);  
    }  
    // name (V) полностью записано.  
    // принимаем в реализацию.  
    int l=0, r=1000000000;  
    while(r-l>1){  
        int mid=(r+l)/2;  
        if bool flag=false;  
        for(int i=mid; i<=n; ++i){  
            vector<vector<bool>> used(n+1, vector<bool>(n+1, false));  
            if (check(used, i, 1, mid) && bfs(i, 1, mid, used)){  
                flag=true;  
                break;  
            }  
        }  
        if (flag){  
            n=mid; l=mid;  
        } else {  
            l=mid; r=mid;  
        }  
    }  
    if (l==0) cout << "-1" << endl; //name нельзя пройти  
    else cout << l << endl;  
    return 0;  
}
```

15+5+1=210

210



Вариант № 4

Задача 5

1) За 1 день одна красная клетка порождает R красных и $(N-R)$ зелёных клеток. Получаем, что за D дней всего красных клеток будет R^D .

2) Посмотрим как изменится количество зелёных клеток. Каждая красная клетка, кроме последнего поколения

код программы на языке Python:

```
N, R, D = map(int, input().split())
```

```
MOD = 1000000000
```

```
MOD = 1000000007
```

```
cnt_r = 1
```

```
cnt_g = 0
```

```
for _ in range(D):
```

```
    cnt_g += (N-R) * cnt_r
```

```
    cnt_r *= R
```

```
    cnt_g %= MOD
```

```
    cnt_r %= MOD
```

```
print(cnt_r, cnt_g)
```

```
print(cnt_r, cnt_g)
```

#* Описание решения: каждый день 1 красная клетка порождает R красных и $(N-R)$ зелёных клеток. Запустим расчёт на D дней, постоянно будем выполнять операцию взятия остатка по модулю. Асимптотика $O(D)$

18+5 = 235

235



Вариант № 7

Задача 4

Помогите найти код ^{цвета} ~~каждого~~ ~~каждого~~ RGB на каждом сегменте рисунков:

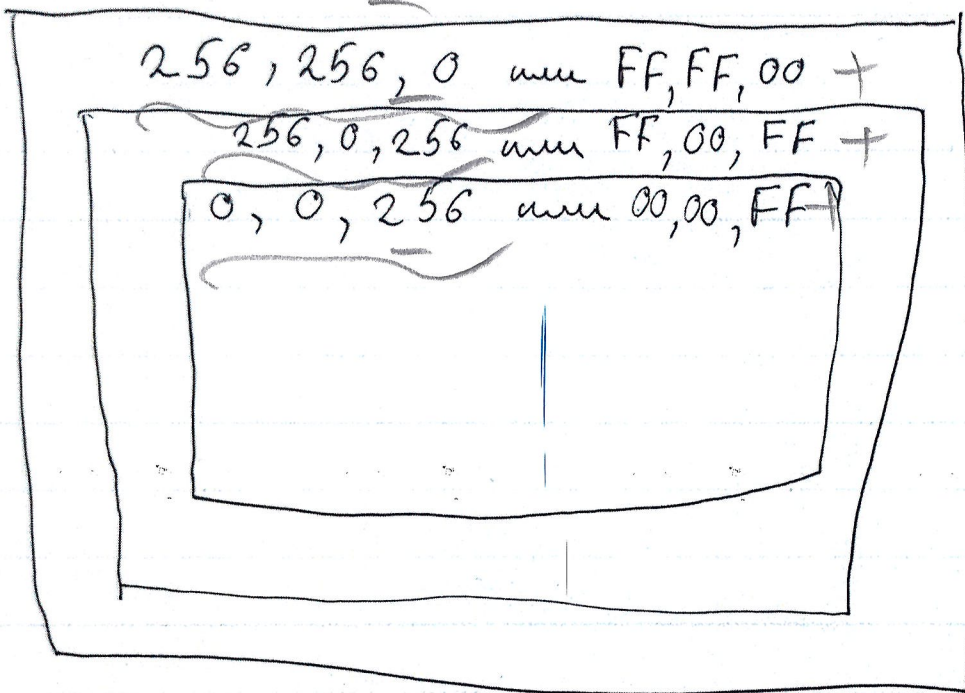


Рисунок 1

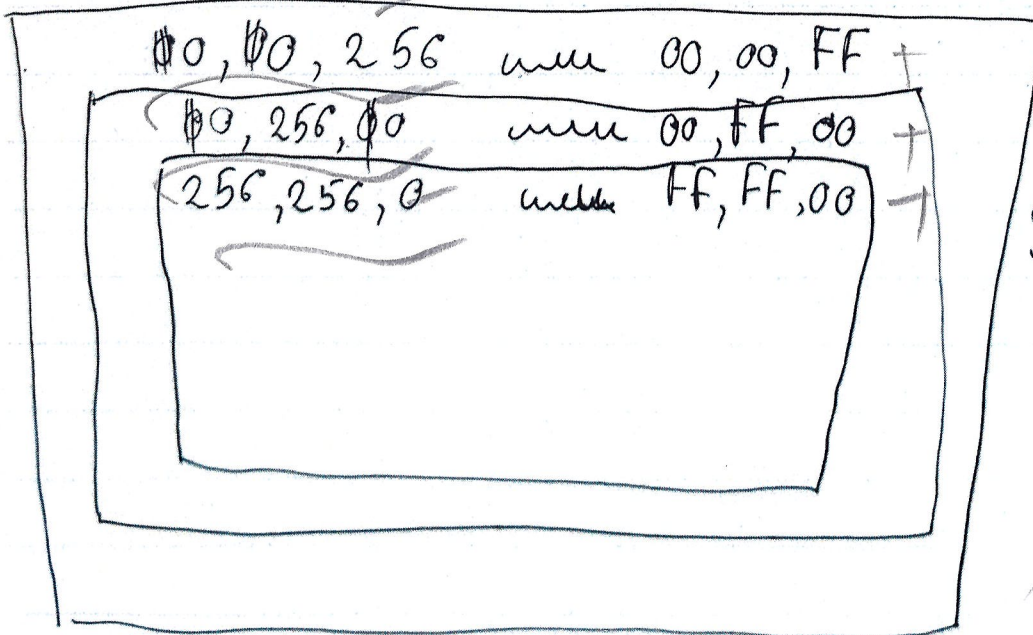
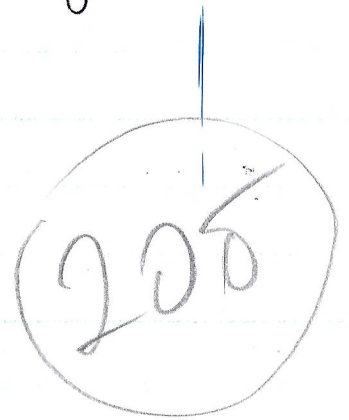
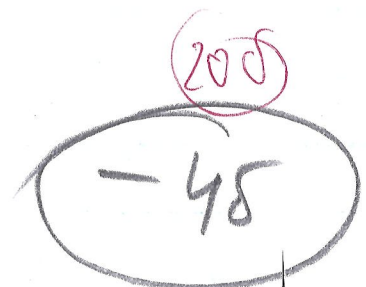


Рисунок 2





Вариант №

7

Получаем, что внешний сегмент равен :
с 256, 256, 0 на 0, 0, 256

средний сегмент:

с 256, 0, 256 на 0, 256, 0

внутренний сегмент:

с 0, 0, 256 на 256, 256, 0

Из чего можно сделать вывод, что процесс
"инверсия" всех всех цветов, а именно, если
(r, g, b) - значения каждого эркости, то
новое значение будет:

$$(256 - r, 256 - g, 256 - b)$$

Ответ: формула алгоритма обработки изобра-
жения: $(256 - r, 256 - g, 256 - b)$, где r, g, b - значения эркости красного, зелёного
и синего каналов соответственно.



Вариант № 7

Задача 3

Перепишем логику

Перепишем логическую схему в виде формулы:

$$F = (x_1 \text{ xor } x_2) \vee (x_2 \& x_3)$$

Составим таблицу истинности:

x_1	x_2	x_3	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1
1	0	0	1
1	0	1	1

Получаем, что $F=1$ при следующих наборах значений (x_1, x_2, x_3) : $(0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 1)$

Преобразуем выражение: $F' = (x_2 \rightarrow x_1) \leftrightarrow (\overline{x_3} \oplus x_1)$

$$(x_2 \rightarrow x_1) \leftrightarrow \overline{x_3} \oplus x_1$$

$$(\overline{x_2} \vee x_1) \leftrightarrow (\overline{x_3} \& \overline{x_1} \vee x_3 \& x_1)$$

Подставим наборы при переменных, при которых $F=1$ в F' :



Вариант №

4

$$(0, 1, 1):$$

$$F' = (1 \vee 1) \leftrightarrow (0 \vee 1), \text{ что истинно}$$

$$(1, 0, 0):$$

$$F' = (0 \vee 0) \leftrightarrow (1 \vee 0), \text{ что истинно}$$

$$(1, 0, 1):$$

$$F' = (0 \vee 0) \leftrightarrow (0 \vee 0), \text{ что истинно}$$

$$(1, 1, 1):$$

$$F' = (0 \vee 1) \leftrightarrow (0 \vee 1), \text{ что истинно}$$

$$(0, 1, 0):$$

$$F' = (1 \vee 1) \leftrightarrow (0 \vee 0), \text{ что ложно}$$

Итак, получаем истинную таблицу:

x_1	x_2	x_3	F	F'	F''
0	0	0	0	—	—
0	0	1	0	—	—
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	0	—	—
1	1	1	1	1	1

Ответ:

$$F' = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$F'' = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

(108) 50%
ошибка

(50)