



You can view this report online at : <https://www.hackerrank.com/x/tests/1132857/candidates/26993979/report>

Email: ramiroSaravia14@gmail.com

Test Name: 2021C1 - 2do Parcial

Taken On: 23 Jun 2021 17:09:54 -03

Time Taken: 162 min 39 sec/ 240 min

Student Roll Number: 641/19

Invited by: Cyntia

Skills Score:

Tags Score:

**100%**  
250/250

scored in **2021C1 - 2do Parcial**  
 in 162 min 39 sec on 23 Jun  
 2021 17:09:54 -03

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	<a href="#">Ejercicio 1 - vecinoRiendose</a> > Coding	42 min 19 sec	100/ 100	✔
Q2	<a href="#">Ejercicio 2 - situacionComica</a> > Coding	8 min 18 sec	50/ 50	✔
Q3	<a href="#">Ejercicio 3 - mayorCantidadPeriodosAnteriores</a> > Coding	1 hour 14 min 12 sec	50/ 50	✔
Q4	<a href="#">Ejercicio 4 - actualizarMapa</a> > Coding	37 min 37 sec	50/ 50	✔

**QUESTION 1**

Correct Answer

---

Score 100

**Ejercicio 1 - vecinoRiendose** > Coding

QUESTION DESCRIPTION

*Ejercicio 1:*

`bool vecinoRiendose(mapa m, posicion p)`

Dado un mapa y una posición devuelve si hay al menos una familia vecina que se está riendo del chiste.

**CANDIDATE ANSWER**

Language used: **C++**

```

1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 typedef pair<int, int> posicion;

```

```

7 typedef int duracionPeriodoActual;
8 typedef int cantidadPeriodosAnteriores;
9 typedef pair<duracionPeriodoActual, cantidadPeriodosAnteriores> estado;
10 typedef vector<vector<estado > > mapa;
11
12
13 bool estaEnRango (mapa m, posicion x){
14     bool res=false;
15     if( (x.first<m.size()&& x.first>=0)&&(x.second>=0&&x.second<m[0].size())){
16         res=true;
17     }
18     return res;
19 }
20
21 bool vecinoRiendose (mapa m, posicion p){
22     bool res=false;
23     if(estaEnRango(m,make_pair(p.first+1,p.second)) && m[p.first+1]
24 [p.second].first>0){
25         res=true;
26     }
27     if(estaEnRango(m,make_pair(p.first-1,p.second)) && m[p.first-1]
28 [p.second].first>0){
29         res=true;
30     }
31     if(estaEnRango(m,make_pair(p.first,p.second+1)) && m[p.first]
32 [p.second+1].first>0){
33         res=true;
34     }
35     if(estaEnRango(m,make_pair(p.first,p.second-1)) && m[p.first][p.second-
36 1].first>0){
37         res=true;
38     }
39     return res;
40 }
41
42
43
44 int main()
45 {
46     /* No hace falta modificar el main */
47
48     // Leo la entrada
49     int n; // Cantidad de filas del mapa
50     int t; // Cantidad columnas del mapa
51     int f; // fila de la posición de la vivienda que quiero ver si tiene
52 vecinos riendose
53     int c; // columna de la posición de la vivienda que quiero ver si tiene
54 vecinos riendose
55
56     cin >> f;
57     cin >> c;
58     cin >> n;
59     cin >> t;
60
61     posicion p = make_pair(f, c);
62
63     string e;
64     int filas = n, columnas = t;
65     estado vacio = make_pair(0, 0);
66     mapa m(filas, vector<estado>(columnas, vacio)); //en m se guarda el mapa
67
68     for (int f = 0; f < filas; ++f) {
69         for (int c = 0; c < columnas; ++c) {
70

```

```

71     cin >> e; //estado de la posición (f,c)
72
73     e = e.substr(1, e.size()-2);
74     int posFound = e.find(',', 0);
75
76     duracionPeriodoActual dpa = stoi(e.substr(0, posFound));
77     cantidadPeriodosAnteriores cpa = stoi(e.substr(posFound+1,
78 e.size()-1));
79
80     m[f][c] = make_pair(dpa, cpa); //guardo el estado en el mapa
81 }
82 }
83
84
85 bool res = vecinoRiendose(m, p);
86
87 //Salida
88 cout << (res ? "true" : "false");
89 return 0;
90 }

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✔ Success	10	0.0151 sec	8.95 KB
Testcase 3	Easy	Hidden case	✔ Success	10	0.015 sec	9 KB
Testcase 5	Easy	Hidden case	✔ Success	10	0.0175 sec	8.95 KB
Testcase 7	Easy	Hidden case	✔ Success	10	0.0169 sec	8.95 KB
Testcase 9	Easy	Hidden case	✔ Success	10	0.0271 sec	8.91 KB
Testcase 11	Easy	Hidden case	✔ Success	10	0.0159 sec	9.01 KB
Testcase 13	Easy	Hidden case	✔ Success	10	0.0165 sec	8.78 KB
Testcase 15	Easy	Hidden case	✔ Success	10	0.0181 sec	8.94 KB
Testcase 17	Easy	Hidden case	✔ Success	10	0.0187 sec	8.96 KB
Testcase 14	Easy	Hidden case	✔ Success	10	0.0157 sec	8.95 KB

No Comments

## QUESTION 2



Correct Answer

Score 50

## Ejercicio 2 - situacionComica > Coding

### QUESTION DESCRIPTION

### Ejercicio 2:

`void situacionComica(mapa m, int &riendo, int &aburridos, int &nuncaEscucharon)`

Dado un mapa queremos saber cuántas familias se están riendo actualmente, cuántas están aburridas (pero ya se rieron del chiste al menos una vez) y cuántas nunca escucharon jamás el chiste.

### CANDIDATE ANSWER

Language used: **C++**

```

1 #include <iostream>
2 #include <vector>
3

```

```

4 using namespace std;
5
6 typedef pair<int, int> posicion;
7 typedef int duracionPeriodoActual;
8 typedef int cantidadPeriodosAnteriores;
9 typedef pair<duracionPeriodoActual, cantidadPeriodosAnteriores> estado;
10 typedef vector<vector<estado > > mapa;
11
12
13 void situacionComica(mapa m, int &riendo, int &aburridos, int
14 &nuncaEscucharon){
15     for(int i=0;i<m.size();i++){
16         for(int j=0;j<m[0].size();j++){
17             if(m[i][j].first==0 && m[i][j].second>0){
18                 aburridos++;
19             }
20             if(m[i][j].first>0){
21                 riendo++;
22             }
23             if(m[i][j].second==0 && m[i][j].first==0){
24                 nuncaEscucharon++;
25             }
26         }
27     }
28 }
29
30
31 int main()
32 {
33     /* No hace falta modificar el main */
34
35     // Leo la entrada
36     int n; // Cantidad de filas del mapa
37     int t; // Cantidad columnas del mapa
38     cin >> n;
39     cin >> t;
40
41     string e;
42     int filas = n, columnas = t;
43     estado vacio = make_pair(0, 0);
44     mapa m(filas, vector<estado>(columnas, vacio)); //en m se guarda el mapa
45
46     for (int f = 0; f < filas; ++f) {
47         for (int c = 0; c < columnas; ++c) {
48             cin >> e; //estado de la posición (f,c)
49
50             e = e.substr(1, e.size()-2);
51             int posFound = e.find(',', 0);
52
53             duracionPeriodoActual dpa = stoi(e.substr(0, posFound));
54             cantidadPeriodosAnteriores cpa = stoi(e.substr(posFound+1,
55 e.size()-1));
56
57             m[f][c] = make_pair(dpa, cpa); //guardo el estado en el mapa
58         }
59     }
60
61
62     int riendo = 0;
63     int aburridos = 0;
64     int nuncaEscucharon = 0;
65     situacionComica(m, riendo, aburridos, nuncaEscucharon);
66
67     //cout << endl;

```

```

67 //Salida
68 cout << riendo << " " << aburridos << " " << nuncaEscucharon;
    return 0;
}

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Hidden case	✔ Success	10	0.0201 sec	9.02 KB
Testcase 1	Easy	Hidden case	✔ Success	10	0.016 sec	8.86 KB
Testcase 2	Easy	Hidden case	✔ Success	10	0.017 sec	8.95 KB
Testcase 3	Easy	Sample case	✔ Success	10	0.0193 sec	8.9 KB
Testcase 4	Easy	Hidden case	✔ Success	10	0.0177 sec	8.88 KB

No Comments

### QUESTION 3



Correct Answer

Score 50

## Ejercicio 3 - mayorCantidadPeriodosAnteriores > Coding

### QUESTION DESCRIPTION

### Ejercicio 3:

`vector<posicion>` mayorCantidadPeriodosAnteriores(`mapa m`)

Dado un mapa devuelve las posiciones de las familias que se rieron durante la mayor cantidad de períodos.

### CANDIDATE ANSWER

Language used: **C++**

```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 typedef pair<int, int> posicion;
8 typedef int duracionPeriodoActual;
9 typedef int cantidadPeriodosAnteriores;
10 typedef pair<duracionPeriodoActual, cantidadPeriodosAnteriores> estado;
11 typedef vector<vector<estado > > mapa;
12
13
14
15 int mayorCantidadPeriodosAnteriores (mapa m){
16     int aux=m[0][0].second;
17     for(int i=0;i<m.size();i++){
18         for(int j=0;j<m[0].size();j++){
19             if(m[i][j].second>=aux){
20                 aux=m[i][j].second;
21             }
22         }
23     }
24     return aux;
25 }
26 vector<posicion> mayorCantidadCarcajadasPasadas (mapa m){
27     vector<posicion> res;
28     int aux=mayorCantidadPeriodosAnteriores(m);

```

```

29     for(int i=0;i<m.size();i++){
30         for(int j=0;j<m[0].size();j++){
31             if(m[i][j].second==aux){
32                 res.push_back(make_pair(i,j));
33             }
34         }
35     }
36     return res;
37 }
38
39
40
41 //Función declarada para que puedan correr los tests.
42 void ordenarSinRepetidos(vector<pair<int,int>> &v);
43
44
45 int main()
46 {
47     /* No hace falta modificar el main */
48
49     // Leo la entrada
50     int n; // Cantidad de filas del mapa
51     int t; // Cantidad columnas del mapa
52
53     cin >> n;
54     cin >> t;
55
56     string e;
57     int filas = n, columnas = t;
58     estado vacio = make_pair(0, 0);
59     mapa m(filas, vector<estado>(columnas, vacio)); //en m se guarda el mapa
60
61     for (int f = 0; f < filas; ++f) {
62         for (int c = 0; c < columnas; ++c) {
63             cin >> e; //estado de la posición (f,c)
64
65             e = e.substr(1, e.size()-2);
66             int posFound = e.find(',', 0);
67
68             duracionPeriodoActual dpa = stoi(e.substr(0, posFound));
69             cantidadPeriodosAnteriores cpa = stoi(e.substr(posFound+1,
70 e.size()-1));
71
72             m[f][c] = make_pair(dpa, cpa); //guardo el estado en el mapa
73         }
74     }
75
76     vector<posicion> res = mayorCantidadCarcajadasPasadas(m);
77
78     //Salida
79     ordenarSinRepetidos(res);
80
81     for(int i = 0; i < res.size(); ++i){
82         cout << "(" << res[i].first << "," << res[i].second << ")" << endl ;
83     }
84     return 0;
85 }
86
87
88
89 /*Esta función no deben modificarla ni usarla para resolver los ejercicios
90 dados.*/
91 void ordenarSinRepetidos(vector<pair<int,int>> &v) {
92     sort(v.begin(), v.end());

```

```

32     sort(v.begin(), v.end());
    v.erase(unique(v.begin(), v.end()), v.end());
}

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Hidden case	✔ Success	10	0.0148 sec	8.87 KB
Testcase 1	Easy	Hidden case	✔ Success	10	0.0146 sec	8.95 KB
Testcase 2	Easy	Sample case	✔ Success	10	0.0136 sec	8.96 KB
Testcase 3	Easy	Hidden case	✔ Success	10	0.0139 sec	9.04 KB
Testcase 4	Easy	Hidden case	✔ Success	10	0.0247 sec	8.89 KB

No Comments

#### QUESTION 4



Correct Answer

Score 50

### Ejercicio 4 - actualizarMapa > Coding

#### QUESTION DESCRIPTION

#### Ejercicio 4:

```
void actualizarMapa(mapa &m)
```

Dado un mapa queremos saber la evolución del chiste por el barrio cuando pasa un día. Se debe actualizar el mapa tomando en cuenta las reglas expresadas en el enunciado.

#### CANDIDATE ANSWER

Language used: C++

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  typedef pair<int, int> posicion;
7  typedef int duracionPeriodoActual;
8  typedef int cantidadPeriodosAnteriores;
9  typedef pair<duracionPeriodoActual, cantidadPeriodosAnteriores> estado;
10 typedef vector<vector<estado > > mapa;
11
12 bool estaEnRango (mapa m, posicion x){
13     bool res=false;
14     if( (x.first<m.size()&&x.first>=0) && (x.second>=0&&x.second<m[0].size()) ){
15         res=true;
16     }
17     return res;
18 }
19
20 bool vecinoRiendose (mapa m, posicion p){
21     bool res=false;
22     if(estaEnRango(m,make_pair(p.first+1,p.second)) && m[p.first+1]
23 [p.second].first>0) {
24         res=true;
25     }
26     if(estaEnRango(m,make_pair(p.first-1,p.second)) && m[p.first-1]

```

```

27 [p.second].first>0) {
28     res=true;
29 }
30     if(estaEnRango(m,make_pair(p.first,p.second+1)) && m[p.first]
31 [p.second+1].first>0) {
32         res=true;
33     }
34     if(estaEnRango(m,make_pair(p.first,p.second-1)) && m[p.first][p.second-
35 1].first>0) {
36         res=true;
37     }
38     return res;
39 }
40
41 void actualizarMapa(mapa &m) {
42     mapa aux=m;
43     for(int i=0;i<m.size();i++){
44         for(int j=0;j<m[0].size();j++){
45             if(aux[i][j].first>0 && aux[i][j].first<=14) {
46                 m[i][j].first++;
47             }
48             if(aux[i][j].first>0 && aux[i][j].first==15) {
49                 m[i][j].second++;
50                 m[i][j].first=0;
51             }
52             if(aux[i][j].first==0 &&
53 vecinoRiendose(aux,make_pair(i,j))==true) {
54                 m[i][j].first++;
55             }
56             if(aux[i][j].first==0 &&
57 vecinoRiendose(aux,make_pair(i,j))==false) {
58                 m[i][j]=m[i][j];
59             }
60         }
61     }
62 }
63
64 int main()
65 {
66     /* No hace falta modificar el main */
67
68     // Leo la entrada
69     int n; // Cantidad de filas del mapa
70     int t; // Cantidad columnas del mapa
71     cin >> n;
72     cin >> t;
73
74     string e;
75     int filas = n, columnas = t;
76     estado vacio = make_pair(0, 0);
77     mapa m(filas, vector<estado>(columnas, vacio)); //en m se guarda el mapa
78
79     for (int f = 0; f < filas; ++f) {
80         for (int c = 0; c < columnas; ++c) {
81
82             cin >> e; //estado de la posición (f,c)
83
84             e = e.substr(1, e.size()-2);
85             int posFound = e.find(',', 0);
86
87             duracionPeriodoActual dpa = stoi(e.substr(0, posFound));
88             cantidadPeriodosAnteriores cpa = stoi(e.substr(posFound+1,
89 e.size()-1));
90

```



```

90     m[f][c] = make_pair(dpa, cpa); //guardo el estado en el mapa
91     }
92 }
93 }
94
95
96 actualizarMapa(m);
97
98 //Salida
99 int cantFilas = m.size();
100 int cantColumnas = m[0].size();
101
102 cout << cantFilas << endl;
103 cout << cantColumnas;
104
105 for (int f = 0; f < cantFilas; ++f) {
106     for (int c = 0; c < cantColumnas; ++c) {
107
108         estado e = m[f][c];
109         if(c == 0){
110             cout <<endl;
111         }
112         cout << "(" << e.first << "," << e.second << ")" << " ";
113
114     }
115 }
116
117 return 0;
118 }

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Medium	Hidden case	✔ Success	10	0.0158 sec	8.91 KB
Testcase 1	Easy	Hidden case	✔ Success	10	0.0127 sec	8.92 KB
Testcase 2	Easy	Hidden case	✔ Success	10	0.0136 sec	9 KB
Testcase 3	Easy	Sample case	✔ Success	10	0.0179 sec	8.78 KB
Testcase 4	Easy	Hidden case	✔ Success	10	0.0177 sec	9.06 KB

No Comments