

Área personal ► Mis cursos ► InfoC++ ► Clase 13 ► Ejemplo EA - 11 (CR)

<b>Comenzado el</b>	lunes, 12 de noviembre de 2018, 21:24
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	jueves, 15 de noviembre de 2018, 19:52
<b>Tiempo empleado</b>	2 días 22 horas
<b>Calificación</b>	Sin calificar aún

## Pregunta 1

Correcta

Puntúa 15 sobre 15

Codifique la definición de una función cuyo prototipo es:

**void agregarBarcos( int tablero[N][N], int nuevos)**

La función recibe como argumentos una matriz NxN de enteros (tablero) y un valor entero (nuevos). La matriz representa un tablero cuadrangular de NxN casilleros. En cada casillero pueden agregarse uno o más barcos. Por lo tanto, el elemento correspondiente a la fila  $x$  y columna  $y$  de la matriz indica la cantidad de barcos posicionados en la coordenada horizontal  $x$  y la coordenada vertical  $y$ . El argumento entero (nuevos) indica la cantidad de nuevos barcos que se deben agregar al tablero. La función debe solicitar que el usuario ingrese tantos pares de coordenadas como indicado por el argumento 'nuevos', y para cada coordenada valida agregar un barco en dicha posición. De existir ya uno o mas barcos en la misma, se debe agregar un nuevo barco incrementando la cantidad preexistente. Si la coordenada ingresada fuera invalida, no debe agregarse barco alguno.

**Respuesta:** (penalty regime: 0 %)

```

1 void agregarBarcos( int tablero[N][N], int nuevos)
2 {
3     int x, y;
4     for(int i=0;i<nuevos;i++)
5     {
6         cin>> x>>y;
7
8         if (x<N && y<N)
9         {
10            tablero[x][y]++;
11        }
12    }
13
14    return;
15 }
```

	Test	Input	Expected	Got	
✓	<pre>int tablero[N][N]={0,0,0,0}; agregarBarcos(tablero,1); for(int i=0; i&lt;N; i++)     for(int j=0; j&lt;N; j++)         cout &lt;&lt; tablero[i][j] &lt;&lt; endl;</pre>	<pre>0 0</pre>	<pre>1 0 0 0</pre>	<pre>1 0 0 0</pre>	✓
✓	<pre>int tablero[N][N]={0,1,0,0}; agregarBarcos(tablero,1); for(int i=0; i&lt;N; i++)     for(int j=0; j&lt;N; j++)         cout &lt;&lt; tablero[i][j] &lt;&lt; endl;</pre>	<pre>0 1</pre>	<pre>0 2 0 0</pre>	<pre>0 2 0 0</pre>	✓

	Test	Input	Expected	Got	
✓	<pre>int tablero[N][N]={0,0,0,0}; agregarBarcos(tablero,2); for(int i=0; i&lt;N; i++)     for(int j=0; j&lt;N; j++)         cout &lt;&lt; tablero[i][j] &lt;&lt; endl;</pre>	0 0 1 1	1 0 0 1	1 0 0 1	✓
✓	<pre>int tablero[N][N]={0,0,0,0}; agregarBarcos(tablero,1); for(int i=0; i&lt;N; i++)     for(int j=0; j&lt;N; j++)         cout &lt;&lt; tablero[i][j] &lt;&lt; endl;</pre>	2 2	0 0 0 0	0 0 0 0	✓
✓	<pre>int tablero[N][N]={0,0,0,0,0,0,0,0,0}; agregarBarcos(tablero,3); for(int i=0; i&lt;N; i++)     for(int j=0; j&lt;N; j++)         cout &lt;&lt; tablero[i][j] &lt;&lt; endl;</pre>	1 0 1 0 2 1	0 0 0 2 0 0 0 1 0	0 0 0 2 0 0 0 1 0	✓

Todas las pruebas superadas. ✓

### Question author's solution:

```
void agregarBarcos(int tablero[N][N], int cantidad)
{
    int x, y;
    for(int i=0; i<cantidad; i++)
    {
        cout << "Ingresar coordenada horizontal (x): ";
        cin >> x;
        cout << "Ingresar coordenada vertical (y): ";
        cin >> y;
        if(x>=0 && x<N && y>=0 && y<N)
            tablero[x][y]++;
    }
}
```

**Correcta**

Puntos para este envío: 15/15.

## Pregunta 2

Correcta

Puntúa 15 sobre 15

Codifique la definición de una función cuyo prototipo es:

**void mostrarTablero( int tablero[N][N])**

La función recibe como argumentos una matriz NxN de enteros que representa un tablero cuadrangular de NxN casilleros. La función de mostrar por pantalla el tablero indicando con el símbolo '-' si no hay barcos en la posición, y con el símbolo 'x' en el caso que haya uno o más barco en la posición.

**Ejemplo:** La matriz {0,1,0,2} debe mostrarse como

-x

-x

**Respuesta:** (penalty regime: 0 %)

```

1 void mostrarTablero( int tablero[N][N])
2 {
3     for(int i=0;i<N;i++)
4     {
5         for(int j=0;j<N;j++)
6         {
7             if(tablero[i][j]==0)
8             {
9                 cout<<'-' ;
10            }
11            else if(tablero[i][j]!=0)
12            {
13                cout<<'x';
14            }
15        }
16        cout<<endl;//completa una fila y tiene que baja
17    }
18    return;
19 }
```

	Test	Expected	Got	
✓	int tablero[N][N]={0,0,0,0}; mostrarTablero(tablero);	-- --	-- --	✓
✓	int tablero[N][N]={1,2,3,4}; mostrarTablero(tablero);	xx xx	xx xx	✓
✓	int tablero[N][N]={1,0,0,0,1,0,0,0,1}; mostrarTablero(tablero);	x-- -x- --x	x-- -x- --x	✓
✓	int tablero[N][N]={0,1,0,0,1,0,0,1,1}; mostrarTablero(tablero);	-x- -x- -xx	-x- -x- -xx	✓

Todas las pruebas superadas. ✓

```
void mostrarTablero(int tablero[N][N])
{
    for(int i=0; i<N; i++)
    {
        for(int j=0; j<N; j++)
            if(tablero[i][j]>0)
                cout<<'x';
            else
                cout<<'-' ;
        cout << endl;
    }
}
```

**Correcta**

Puntos para este envío: 15/15.

## Pregunta 3

Correcta

Puntúa 15 sobre 15

Codifique la definición de una función cuyo prototipo es:

**int cantidadFlotas( int tablero[N][N], int umbral)**

La función recibe como argumentos una matriz NxN de enteros que representa un tablero cuadrangular de NxN casilleros donde cada elemento indica la cantidad de barcos en cada casillero, y un valor entero (umbral) que indica la cantidad mínima de barcos en una misma posición que forman una flota. La función debe devolver la cantidad de flotas presentes en el tablero, considerando una flota toda aquella cantidad que cumpla con el mínimo de barcos indicado por el argumento umbral.

**Ejemplo:** Para la matriz {0,1,0,2} con umbral=2 (una flota debe tener al menos dos barcos), la función debe devolver el valor 1.

**Respuesta:** (penalty regime: 0 %)

```

1  int cantidadFlotas( int tablero[N][N], int umbral)
2  {
3
4      int cuenta=0;
5      for(int i=0;i<N;i++)
6      {
7          for(int j=0;j<N;j++)
8          {
9              if(tablero[i][j]>=umbral)
10             {
11                 cuenta++;
12             }
13         }
14     }
15     return cuenta;
16 }
```

	Test	Expected	Got	
✓	int tablero[N][N]={0,0,0,0}; cout << cantidadFlotas(tablero,0);	4	4	✓
✓	int tablero[N][N]={0,0,0,0}; cout << cantidadFlotas(tablero,1);	0	0	✓
✓	int tablero[N][N]={0,1,1,0}; cout << cantidadFlotas(tablero,1);	2	2	✓
✓	int tablero[N][N]={0,0,2,0,1,0,0,3,0}; cout << cantidadFlotas(tablero,2);	2	2	✓
✓	int tablero[N][N]={0,0,2,0,1,0,0,3,0}; cout << cantidadFlotas(tablero,3);	1	1	✓

Todas las pruebas superadas. ✓

```
int cantidadFlotas(int tablero[N][N], int umbral)
{
    int suma = 0;
    for(int i=0; i<N; i++)
        for(int j=0; j<N; j++)
            if(tablero[i][j]>=umbral)
                suma++;

    return suma;
}
```

**Correcta**

Puntos para este envío: 15/15.

## Pregunta 4

Correcta

Puntúa 1 sobre 1

Codifique la definición de una función cuyo prototipo es:

**int dispararBarcos( int tablero[N][N], int x, int y)**

La función recibe como argumentos una matriz NxN de enteros que representa un tablero cuadrangular de NxN casilleros donde cada elemento indica la cantidad de barcos en un casillero, un par de coordenadas (x,y) que indican la posición del disparo. En caso que en esta posición existan barcos, la función debe actualizar el elemento de la matriz con cero (todos los barcos hundidos), y devolver la cantidad de barcos que había en dicha posición y fueron hundidos.

**Ejemplo:** Para la matriz {0,1,0,2} y un disparo en (1,1) la función debe devolver el valor 2.

**Respuesta:** (penalty regime: 0 %)

```

1  int dispararBarcos( int tablero[N][N], int x, int y)
2  {
3      int hundidos;
4
5      if(tablero[x][y]!=0)
6      {
7          hundidos=tablero[x][y];
8          tablero[x][y]=0;
9      }
10
11     return hundidos;
12 }
```

	Test	Expected	Got	
✓	int tablero[N][N]={0,0,0,0}; cout << dispararBarcos(tablero,1,0);	0	0	✓
✓	int tablero[N][N]={0,1,0,0}; cout << dispararBarcos(tablero,0,1);	1	1	✓
✓	int tablero[N][N]={0,1,1,0}; cout << dispararBarcos(tablero,0,0);	0	0	✓
✓	int tablero[N][N]={0,1,1,0,12,0,0,1,0}; cout << dispararBarcos(tablero,1,1);	12	12	✓

Todas las pruebas superadas. ✓

**Question author's solution:**



```
int dispararBarcos(int tablero[N][N], int x, int y)
{
    int resultado = tablero[x][y];
    if(tablero[x][y]>0)
        tablero[x][y]=0;
    return resultado;
}
```

**Correcta**

Puntos para este envío: 1/1.

## Pregunta 5

Finalizado

Puntúa como 1

Realizar un programa en C++ que:

- 1) Considere un tablero de 3x3 casillas. Dicho tablero debe representarse a través de una matriz entera, donde inicialmente cada elemento sea cero (indicando que no hay barcos)
- 2) Solicite al usuario que ingrese por teclado la cantidad de barcos que desea agregar al tablero. Ver ejemplo de ejecución al final del enunciado.
- 3) Agregue la cantidad de barcos indicada utilizando la función agregarBarcos(). Ver ejemplo de ejecución al final del enunciado.
- 4) Muestre por pantalla el tablero con las posiciones donde hay barcos presentes utilizando la función mostrarTablero(). Ver ejemplo de ejecución al final del enunciado.
- 5) Muestre por pantalla la cantidad de flotas que contienen al menos dos barcos en la misma posición, utilizando la función cantidadFlotas(). Ver ejemplo de ejecución al final del enunciado.
- 6) Solicite al usuario ingresar las coordenadas (x,y) de tantos disparos como flotas haya en el tablero. Para cada coordenada ingresada se deberá verificar que el disparo sea válido (es decir, la coordenada se encuentre dentro del tablero), caso contrario, se debe mostrar un error y pedir se ingrese nuevamente su valor hasta que la misma sea válida. Ver ejemplo de ejecución al final del enunciado.
- 7) Una vez realizado todos los disparos, se debe mostrar por pantalla la cantidad de barcos hundidos. Ver ejemplo de ejecución al final del enunciado.
- 8) Finalmente, se debe mostrar por pantalla el tablero con las posiciones donde hay barcos presentes utilizando la función mostrarTablero(). Ver ejemplo de ejecución al final del enunciado.

**Ejemplo (N=3):**

Barcos a agregar: 5

Ingresar coordenada horizontal (x): 0

Ingresar coordenada vertical (y): 0

Ingresar coordenada horizontal (x): 1

Ingresar coordenada vertical (y): 0

Ingresar coordenada horizontal (x): 0

Ingresar coordenada vertical (y): 1

Ingresar coordenada horizontal (x): 0

Ingresar coordenada vertical (y): 0

Ingresar coordenada horizontal (x): 3

Ingresar coordenada vertical (y): 3

xx-

x--

---

Flotas dentro del tablero: 2

Ingresar coordenada horizontal (x): 0

Ingresar coordenada vertical (y): 0

Ingresar coordenada horizontal (x): 1

Ingresar coordenada vertical (y): 3

Error: Ingresar nuevamente coordenada vertical (y): 0

Barcos hundidos: 3

xx-

---

---

```
#include <iostream>
using namespace std;

const int N=3;

void agregarBarcos( int tablero[N][N], int nuevos)
{
    int x, y;
    for(int i=0;i<nuevos;i++)
    {
        cout<<"Ingresar coordenada horizontal (x): ";
        cin>>x;
        cout<<"Ingresar coordenada vertical (y): ";
        cin>> y;
```

```
#include<iostream>
using namespace std;
const int N = 2;
// Prototipos de funciones
.....

int main()
{
    int miTablero[N][N];
    int cantBarcos;
    for(int i=0; i<N; i++)
        for(int j=0; j<N; j++)
            miTablero[i][j]=0;

    cout << "Barcos a agregar: ";
    cin >> cantBarcos;
    agregarBarcos(miTablero, cantBarcos);
    mostrarTablero(miTablero);
    int flotas = cantidadFlotas(miTablero,2) ;
    cout << "Flotas dentro del tablero: " << flotas << endl;

    int x, y, hundidos=0;
    for(int i=0;i<cantBarcos;i++)
    {
        cout << "Ingresar coordenada horizontal (x): ";
        cin >> x;
        while(x<0 || x >=N)
        {
            cout << "Error: Ingresar nuevamente coordenada horizontal (x): ";
            cin >> x;
        }
        cout << "Ingresar coordenada vertical (y): ";
        cin >> y;
        while(y<0 || y >=N)
        {
```

```
cout << "Error: Ingresar nuevamente la posición de un barco." << endl;
cin >> y;
}

hundidos+=dispararBarcos(miTablero, x, y);

}
cout << "Barcos hundidos: " << hundidos;
}

// Definiciones de funciones
.....
```