socrative
A showbie PRODUCT

Nombre _____
Fecha _____

Modelamiento_I

Puntuación _____

1. Un problema de optimización se caracteriza por tener lo siguiente:

(A) Parámetros de entrada, Variables de decisión, Restricciones y Función objetivo.

(B) Variables de decisión, Restricciones y Función objetivo.

(C) Parámetros de entrada, Variables de decisión, Restricciones.

2. Al modificar un problema de tal manera que algunos de sus valores constantes se tornen en parámetros, se está:

(A) Generalizando el problema.

(B) Instanciando el problema.

(C) Modificando el modelo, pero el problema sigue siendo exactamente el mismo.

3. Un problema de satisfacción por restricciones no cuenta con:

(A) Parámetros de entrada

(B) Función objetivo

(C) Restricciones

4. Para el problema mochila, indique cual de las siguientes opciones no puede corresponder a un parámetro de entrada:

(A) Número de objetos

(B) Capacidad de la mochila

(C) Peso de cada objeto

(D) $x_i = 1$, si el objeto$_i$ es seleccionado para ser guardado en la mochila y 0 en caso contrario.

5. Para el problema de coloreado de mapas, indique cual de las siguientes opciones no puede corresponder a un parámetro de entrada:

(A) Número de colores a utilizar.

(B) Número de territorios a colorear.

(C) El color de cada territorio.

(D) La información de vecindad entre los diferentes territorios.

6. Considerando el problema de coloreado de mapas sobre la imagen, determine con que valores de nc habría solución:

(A) nc = 1

(B) nc= 4

(C) nc = 2

(D) nc = 3

(E) nc = 9

7.

```
array[1..nt] of var 1..nc: Coloreado;
```

En la instrucción en la imagen, se definen cuantas variables de decisión:

(A) 1

(B) nt

(C) nc

(D) 0

8.

```
constraint forall(i in 1..numFilas) (Coloreado[
    vecinos[i,1] ] != Coloreado[ vecinos[i,2] ]);
```

En la instrucción de la imagen, se definen cuantas restricciones:

(A) 2*numFilas

(B) 2

(C) i

(D) numFilas

9. La restricción $$\sum_{i\mathop{=}1}^n_{}peso_i\cdot x_i<=capacidad_{}-mochila$$ se podría modelar en minizinc como:

(A) constraint sum(i in 1..n)(peso[i]* x[i]) <= capacidad-mochila

(B) constraint forall(i in 1..n) (peso[i] *x[i] <= capacidad-mochila)

(C) no se puede modelar en minizinc.