

# CIENCIA DE LA COMPUTACIO´ N

SIMULACIÓN DE POSIBLES RESULTADOS DE LOS PARTIDOS

# CIENCIA DE LA COMPUTACIO´ N I-CComp 2-1

SANDRO CÁCERES OLAVE RENZO GONZALES CARBAJAL

ALVARO CERNA RAMOS 1ER SEMESTRE

2017

“Los alumnos declaran haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo”

1. **OBJETIVO GENERAL:**

Generar una tabla de posiciones de las eliminatorias RUSIA-2018 dando a conocer todas las posibilidades de los resultados en la FECHA 16 (MARTES 05 de Septiembre). Teniendo en cuenta los siguientes datos:

* Brasil 36 pts
* Colombia 25 pts
* Uruguay 24 pts
* Chile 23 pts
* Argentina 23 pts
* Perú 21 pts
* Paraguay 21 pts
* Ecuador 20 pts
* Bolivia 10 pts
* Venezuela 07 pts

GANA = 3 pts más - PIERDE = 0 pts más - EMPATA = 1 pt más para los dos

1. **CONTENIDO:**

En esta actividad realizamos:

Permutaciones, listas, almacenamiento de datos, parámetros, bucles, algoritmo de ordenamiento, funciones.

1. **MÉTODO:**

Este problema no se puede usar combinatoria porque si importa el orden, por lo cual debemos usar permutaciones donde este orden sí importa y admite repeticiones.

Los cálculos para las permutaciones son el número de elementos que hay elevado al número de elementos que se deben seleccionar.

AHORA LO IMAGINAREMOS COMO UN CANDADO DE CLAVE…

Este tipo de candados se tienen ruedas enumeradas del 1 al 9 donde podrás formar números como el 000 hasta el 999, pero en nuestro caso imaginemos que tiene las ruedas enumeradas del 0 al 2. Ahora tengo solo tres elementos que son GANÓ, EMPATÓ o PERDIÓ, y tengo que elegirlos 5 veces (Ya que en este problema tan específico juegan 10 equipos lo que nos da 5 partidos, nosotros tomamos solo la mitad de equipos que son 5 ya que si uno GANA el otro automáticamente PIERDE, y si uno EMPATA el otro también), entonces mis posibilidades son 3 a la 5ta potencia que serían 243.

Ahora para calcular nuestras 3 opciones, le asignamos a cada una un número:

0 PERDIÓ

1 EMPATÓ

2 GANÓ

Si nos damos cuenta esto sucede en base 3 entonces, lo que hacemos para calcular todas nuestras posibilidades es contar del 0 al 243 en base 3, donde si nos fijamos bien es como el ejemplo del candado cuando no sabes la clave puedes probar todas las combinaciones, si tiene 3 ruedas pruebas todas las combinaciones del 000 al 999, aquí sucede lo mismo solo que en vez de tener 3 ruedas tendremos 5 ruedas que están enumeradas del 0 al 2, donde podremos probar todas las combinaciones desde el 00000 al 22222 contando en base 3.

Finalmente sumamos los puntos a quienes correspondan, luego se ordena en una tabla de acuerdo al puntaje e imprimimos. Observando el resultado nos daremos cuenta que el programa imprime todas las posibilidades que tiene un equipo para poder clasificarse y también nos imprime quienes perdieron, ganaron o empataron, haciendo al código muy útil a la hora de probar posibilidades, ya que sus resultados se ajustan a cualquier evento que pueda pasar en los partidos.

1. **DESARROLLO:**

Inicializamos haciendo una permutación previa la cual nos daba como resultado de 0 a 243 combinaciones posibles dándonos cuenta que trabajaríamos en un número de base 3, creando así el primero algoritmo el cual consiste convertir todo a base 3, muy aparte que lo va almacenando en una lista.

Es así que creamos la función que nos permitirá mostrar la tabla de posiciones muy aparte que decidimos definir a cada equipo como un número.

Sucesivamente creamos el int main()(principal) donde creamos una lista que almacenara el número convertido a base 3, creamos también dos listas una almacenando los puntos en la tabla pero en el orden de los partidos para darnos una ventaja de trabajar con ellos fácilmente y otra la cual almacene el número respectivo de cada equipo, estas dos dentro de un bucle que va a relacionarlas entre sí en un numero de 0 a 243 que es el máximo de permutaciones.

Siguiendo con el razonamiento, a lo que queremos llegar es una lista ordenada siendo así que creamos el algoritmo de ordenamiento InsertionSort el cual consiste en ir comparando la lista y si hay un número que no corresponde entonces lo colocara donde debería ser.

Luego creamos una función que imprima los resultados que tendrían que darse para generar la tabla, además damos los parámetros que define 0 como pierde, 1 como empata y 2 como gana, creamos simultáneamente una función la cual sería indicar si en la lista hay un 0 agrega 3 puntos al rival, si en la lista hay 1 agrega un punto a cada 1, si en la lista hay 2 agrega 3 puntos al equipo anfitrión.

Para terminar en el int main()(principal) ejecutamos todas las funciones creadas el cual quedaría asi:

**int main()**

{

**//lista donde se almacenara el número convertido a base 3**

int lista[5]={0,0,0,0,0};

**//Bolivia,Chile,Ecuador,Peru,Paraguay,Uruguay,Colombia,Brasil,Argentina,Venezuela**

**//for que recorre todos los números del 0 al 243, que son las posibilades que hay**

**//aqui se ejecutan las diferentes funciones**

for(int x=0;x<243;x++){

**//array que almacena los puntos en la tabla en orden de partidos**

int puntos[10]={10,23,20,21,21,24,25,36,23,7};

**//array que almacena los equipos que son representados por un numero**

int equipos[10]={0,1,2,3,4,5,6,7,8,9};

**//convertir a base 3**

convertToBase3(x,lista);

**//añadir puntos**

addPoints(lista,puntos);

***//ordenar las listas***

insertionSort(equipos,puntos);

**//imprimir la tabla de posiciones**

printTable(equipos,puntos,x);

**//indicar los resultados que deben darse para generar la tabla**

printResults(lista,equipos);

}