Nombre: CRISTIAN SULBARAN

DNI: 95943678

**PRUEBA TÉCNICA**

TEST DE PROGRAMACIÓN.

Problema #1. Explique cómo encontrar el mayor elemento de un array

de enteros:

    Ej. (5 8 0 -10 44 89 1 3 7 77 12 -3 4)

    Debería devolver 89.

    Asumiendo que el argumento que se introduce en el algoritmo es

    estrictamente un array de números enteros.

*function* mayorEntero (*arr*) {

*let* arrOrdenado = *arr*.sort((*a*, *b*) *=>* *b* - *a*)

    mayorEntero = arrOrdenado.shift();

    return mayorEntero;

};

* Se aplica una función al array que nos ordene, de mayor a menor, los valores del array
* Se extrae el primer valor resultante de la operación previa (el número mayor).
* Se retorna el valor extraído previamente

console.log(mayorEntero([5, 8, 0, -10, 44, 89, 1, 3, 7, 77, 12, -3, 4]));

LOG => 89

En el caso de que se vaya a pasar cualquier tipo de dato como argumento en el algoritmo, se procede a ordenar más instrucciones, que filtren los

datos y se puedan dar los resultados adecuados:

*function* mayorEnteroFiltro (*posibleArr*) {

    if(!*Array*.isArray(*posibleArr*)) {

     console.log("El argumento no es un array");

    } else {

*let* arrFiltrado = *posibleArr*.filter( *valor* *=>* *Number*.isInteger(*valor*));

        console.log("El número entero mayor del array suministrado es: " + arrFiltrado.sort( (*a*, *b*) *=>* *b* - *a*).shift());

        return "El número entero mayor del array suministrado es: " + arrFiltrado.sort( (*a*, *b*) *=>* *b* - *a*).shift();

    }

};

* Se evalúa si el argumento no es un array para detener el proceso al principio.  Por lo que devuelve un mensaje aclarando el asunto
* Si no es el caso, y en efecto es un array, se utiliza una función  que itere y filtre cada elemento de un array, mediante una función se confirma si dicho elemento es un número entero o no.
* El valor retornado será el primer elemento del array, después de haber sido ordenado de mayor a menor todos los valores filtrados dentro del array.

//mayorEnteroFiltro([5, 8, 0, -10, 44, 89, 1, 3, 7, 77, 12, -3, 4]);

LOG => 89

Se puede utilizar funciones implícitas en el lenguaje de programación, que realicen éste tipo de operaciones, de manera más efectiva,

para agilizar procesos también.

*function* mayorEnteroSimple (*arr*) {

    return Math.max(...*arr*);

}

mayorEnteroSimple([5, 8, 0, -10, 44, 89, 1, 3, 7, 77, 12, -3, 4]);

LOG => 89

PROBLEMA #2: DADO UN ARRAY DE ENTEROS, HACER UNA FUNCIÓN QUE INDIQUE QUÉ RANGOS DE NÚMEROS CONSECUTIVOS

DAN COMO RESULTADO LA CANTIDAD QUE SE LE PASA COMO PARÁMETRO

Ej. array = [6, 7, 5, 4, 3, 1, 2, 3, 5, 6, 7, 9, 0, 0, 1, 2, 4, 1, 2, 3, 5, 1, 2]

sum = 13;

Elementos entre (0 , 1) == 13

Elementos entre (2 , 5) == 13

Elementos entre (3 , 7) == 13

Elementos entre (9 , 10) == 13

Elementos entre (12 , 19) == 13

Elementos entre (13 , 19) == 13

Elementos entre (14 , 19) == 13

Elementos entre (18 , 22) == 13

*function* sumaRangos(*arr*, *sum*) {

*let* i = 0, b = i + 1, c = *arr*[i];

      while (i < *arr*.length && typeof *arr*[i] === "number") {

          if(typeof *arr*[b] === "number") {

              c += *arr*[b];

                 if(c === *sum*) {

                  console.log('La suma de números dentro del rango: (' + i + ', ' + b + ') = ' + *sum* );

                     i++;

                     b = i + 1;

                     c = *arr*[i];

                 } else if( c > *sum* ) {

                 i++;

                     b = i + 1;

                     c = *arr*[i];

                 } else if( c < *sum* ) {

                     b++;

                 }

             }

        }

};

1. Tomo los dos argumentos que nos dan (array, num)
2. Se crea una variable para tomar como índice iterador (i) sobre el array, una variable para utilizar como índice mutable (b) para diferentes alcances dentro del array, y otra para empezar a analizar que elementos del array serán los que pertenecen al rango, y sumen entre si un número igual al del argumento "sum" (c).
3. Un iterador ‘while’(iterador que mientras se cumpla una condición se seguirá cumpliendo las condiciones que lo integren, y termine el ciclo una vez analizado todos los datos correspondientes)

Que mientras se cumplan las funciones en la que el elemento  iterador no sea mayor a la longitud del array y pueda seguir cada elemento dentro del array sin dificultad, y una segunda condición  que exija solo valores numéricos dentro del array, y así termine de iterar sin dificultad de tomar algún elemento no definido dentro o fuera del array, para evitar errores.

1. Si se cumple la condición de que el segundo valor a tomar para la  realización de la suma es un número y no un valor no definido, se  podrá continuar y terminar el loop sin problemas. Por lo tanto la  suma ocurre.

Además se verificará a continuación si en efecto la suma dentro del rango de números da como total un número igual, mayor o menor al  número del argumento "num".

1. Si en efecto la suma de los números dentro del rango es igual al  número del argumento "num", se dará a saber de que número a qué  número, es el rango de elementos el cuál la suma de sus elementos  es igual al número del argumento "num",para pasar al siguiente  valor del array y seguir analizando rangos, se aumenta por 1 el  número del iterador.
2. Además se vuelven a actualizar los datos de las variables para  realizar las mismas operaciones, pero con el número iterador  aumentado, dando paso a los siguientes datos dentro del array
3. En dado caso de que el rango en que se esté analizando la suma de  sus elementos, es mayor al valor del argumento "num", se aumentará el número del iterador por uno para seguir analizando nuevos rangos.
4. Además se actualizarán los datos de las variables nuevamente para comenzar a analizar los datos de un nuevo rango con datos diferentes.
5. Si se presenta el caso de que la primera suma de números dentro del rango desde dónde nos indique el número iterador del  array, sea menor al valor del argumento "num",se aumenta +1 el número mutable (b), para darle continuidad al iterador y avanzar con el siguiente  número, así determinar si la suma de éste junto con los valores

previamente sumados, queda dentro del rango de números aceptable, y que dicha suma sea igual al valor "num". En ésta condición se  evalúa si se sigue sumando el siguiente valor dentro del array,  para alcanzar dicho valor "num"; las demás condiciones evaluarán si en definitiva es igual y se muestre el rango, o en su defecto sea mayor y se sigan analizando los demás valores siguientes dentro del array.

sumaRangos([6, 7, 5, 4, 3, 1, 2, 3, 5, 6, 7, 9, 0, 0, 1, 2, 4, 1, 2, 3, 5, 1, 2], 13);

/\* LOG => ( La suma de números dentro del rango: (0, 1) = 13

La suma de números dentro del rango: (2, 5) = 13

La suma de números dentro del rango: (3, 7) = 13

La suma de números dentro del rango: (9, 10) = 13

La suma de números dentro del rango: (12, 19) = 13

La suma de números dentro del rango: (13, 19) = 13

La suma de números dentro del rango: (14, 19) = 13

La suma de números dentro del rango: (18, 22) = 13 ) \*/

PROBLEMA #3: DADA UNA LISTA SENCILLAMENTE ENLAZADA, EXPLICAR COMO OBTENER EL ELEMENTO DEL MEDIO. EN CASO DE TENER UNA CANTIDAD DE ELEMENTOS PAR,  PUEDE SER CUALQUIERA DE LOS DOS. SI USA UN MÉTODO QUE DEVUELVE EL TAMAÑO  DE LA LISTA, DEBE DEFINIRLO

    Ej. Árbol->Casa->Ruta           Pedro->Juan->José->Marcos

    R= Casa                         R= Juan o José, ambos válidos

Asumiendo una lista sencilla que tiene un elemento en el medio de la  lista al menos,tendrá un elemento al inicio, uno al medio y otro al final, o sea indices 0, 1 y 2, se procede a extraer el elemento de la lista con índice 1.

*function* elemento(*arr*) {

    console.log(*arr*[1]);

    return *arr*[1]

}

elemento(["Arbol", "Casa", "Ruta"]);

LOG => Casa

Si se tiene una lista, en la cuál se requiere extraer 1 o más elementos  que no se ubiquen ni al principio ni al final, solo los del medio, se  procede a crear el algoritmo que retire los elementos del principio y el final, retornando solo los elementos del medio restantes.

*function* elementosDelMedio(*arr*) {

*let* newArr = *arr*;

    newArr.shift();

    newArr.pop();

    console.log(newArr.join(", "));

    return newArr.join(", ");

};

elementosDelMedio(["Pedro", "Juan", "José", "Marcos"]);

LOG => Juan, José

1. Asignarle el array a una nueva variable, para no modificar el array original.
2. Usar una función que retire el primer elemento de la lista.
3. Luego una función que retire el último, para así solo dejar los  elementos del medio en el array.
4. Se devuelve los elementos del medio, retirándolos del array, y separados por una coma para retornarlos como una lista sencillamente  enlazada.

elementosDelMedio(["Pedro", "Juan", "José", "Marcos"]);

LOG => Juan, José

Función que evalúa la longitud de la lista para devolver un valor según sea el caso, se aplicará la función para arrays y string ".length", que mide la longitud del elemento.

*function* longitudListaElementos(*arr*) {

*let* newArr = *arr*;

    if(newArr.length < 3) {

        console.log("No tiene elemento en el medio");

    } else if (newArr.length === 3){

        console.log("El elemento del medio es: " + newArr[1]);

        return newArr[1];

    } else {

        console.log("Los elementos del medio son: " + newArr.splice(1,newArr.length-2) + ".");

        return newArr.splice(1,newArr.length-2);

    }

1. Función que evalúa la longitud de la lista para devolver un valor  según sea el caso, se aplicará la función para arrays y string  ".length", que mide la longitud del elemento.
2. Se asigna el array a una nueva variable para no modificar el array original.
3. Condición que verifica si un elemento posee menos de 3 elementos,  si es así, no posee elemento en el medio.
4. Condición que evalúa que la longitud tiene 3 elementos, por lo que posee al menos un elemento medio.
5. A partir de la segunda condición tendrá más elementos en el medio, y se procede a devolverlos todos.

longitudListaElementos(["Pedro", "Juan", "José", "Marcos", "Perez", "Jimenez"]);

LOG => Los elementos del medio son: Juan,José,Marcos,Perez.

PROBLEMA #4: UN PALÍNDROMO ES UNA PALABRA O FRASE QUE SE LEE IGUAL AL DERECHO O AL REVÉS. HACER UNA FUNCIÓN QUE DETERMINE SI UNA PALABRA O FRASE ES UN PALÍNDROMO

    Ej. es\_palindromo("NEUQUEN") => TRUE

        es\_palindromo("SANTA FE") => FALSE

Si se requiere chequear si una palabra(sin apóstrofes o caractéres especiales) es un palíndromo:

*function* palabra\_palindromo(*str*) {

*let* newArr = *str*.split("").reverse();

        if( newArr.join("") === *str* ) {

            console.log( 'Sí es un palíndromo' );

        } else {

            console.log( 'No es un palíndromo' );

        }

    }

1. Para utilizar una función que invierta el orden de cada letra de un array 'reverse()', y asi chequear si la palabra invertida se lee  igual a la palabra original, se separa cada letra del string dentro de un array con la función 'split("")' .
2. La condición para que devuelva si es un palindromo, es que la  palabra original se lea igual a la palabra invertida previamente.
3. Si no se cumple la condición anterior, no es una palabra palíndromo.

palabra\_palindromo("alomomola");

LOG => 'Sí es un palíndromo'

Si se trata de una palabra con caractéres especiales, o frases(que

incluyen espacios para separar cada palabra) se determinan filtros

para retirarlos y solo evaluar las letras del argumento a evaluar.

*function* es\_palindromo(*str*) {

*let* firstStr = *str*;

*let* oneRegEx = /[a-z0-9]/ig;

*let* filteredStr = firstStr.toLowerCase().match(oneRegEx);

*let* arrToStr1 = filteredStr.join(""),

            arrToStr2 = filteredStr.reverse().join("");

        if (arrToStr2 === arrToStr1) {

            console.log('Sí es un palíndromo');

            return true;

        } else {

            console.log('No es un palíndromo');

            return false;

        }

    }

1. Se copia el argumento a una nueva variable para invertir.
2. Se crea un filtro, para poder obtener solo las letras y números.
3. Se aplica el filtro y se extraen solo las letras(convertidas a

minúsculas) y números para convertirlos en dos casos que se

compararan para saber si se trata o no de un palíndromo.

1. Caso 1 unión de palabra o frase sin espacios ni caractéres

especiales.

1. Caso 2 unión de palabra o frase sin espacios ni caractéres

especiales una vez haya terminado de ser revertido su orden.

1. Se hace la comparación entre los dos casos en esta condición.
2. Si caso 1(frase o palabra normal) comparado con caso 2 (palabra

o frase revertida) son iguales, y por lo tanto un palíndromo.

1. Si no se cumple la primera condición, quiere decir que no

son iguales el caso 1 y el caso 2, por lo que no son palíndromos.

    es\_palindromo("anita lava LA TiNa");

LOG => 'Sí es un palíndromo'

EVALUACIÓN DE UNIX / LINUX.

1. ¿Con qué comando se averigua el directorio actual?

-$ pwd -P

1. ¿A qué directorio espera que se cambie si se ejecuta ‘cd’ sin argumentos?

-Al directorio principal (home)

3- ¿Cómo se obtiene un listado extendido de archivos (con tamaño, fecha etc.)?

-$ ls –l

4- ¿ Qué es un proceso? ¿Cómo obtengo una lista de todos los procesos de mi usuario?

- Un proceso es una instancia de un programa en ejecución. Programas y procesos son entidades distintas. En un sistema operativo multitarea, múltiples instancias de un programa pueden ejecutarse simultáneamente. Cada instancia es un proceso separado.

Prácticamente todo lo que se está ejecutando en el sistema en cualquier momento es un proceso, incluyendo la shell, el ambiente gráfico, que puede tener múltiples procesos, etc.

(<https://bioinf.comav.upv.es/courses/unix/control_procesos.html> )

$ps –u “usuario” (<https://dimitriacosta.xyz/listar-procesos-y-mostrar/> )

Dada la siguiente salida :

>/bin/ls –a

. .. . harry Hat boat hill house push tree

5- ¿Qué imprime el comando “echo h\*”?

-harry hill house

6- Estando en el mismo directorio que en la pregunta 5 ¿Qué valor tendría “$1” si se llama al siguiente script con “h\*” como argumento?

#/bin/ksh

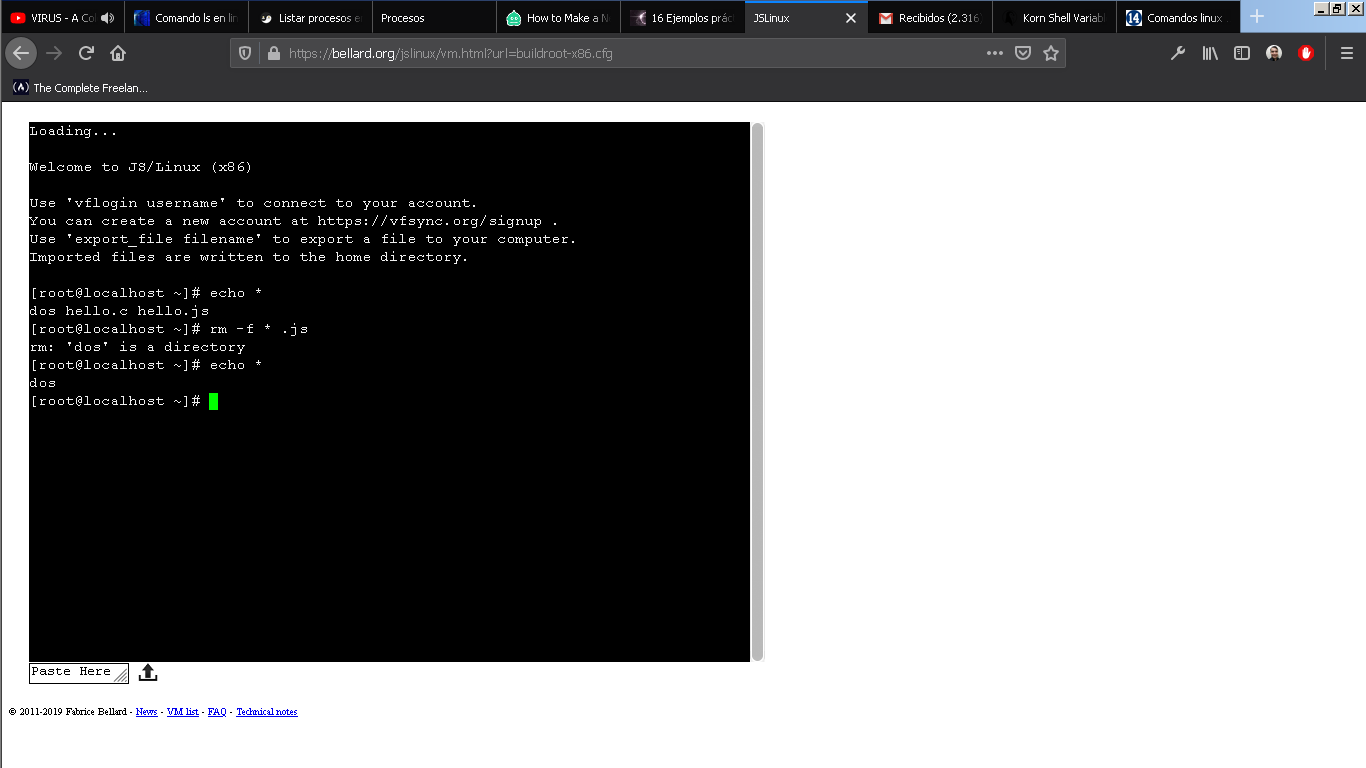
echo my first argument is $1

1. h\*
2. .harry
3. Hill
4. Hat
5. Ninguno de los anteriores

-e. Ninguno de los anteriores

7- ¿Qué ocurre si por accidente ingresa un espacio extra al querer borrar todos los archivos con extensión “.old” y escribe “rm –f \* .old”?

- Me removió todos los archivos excepto los directorios

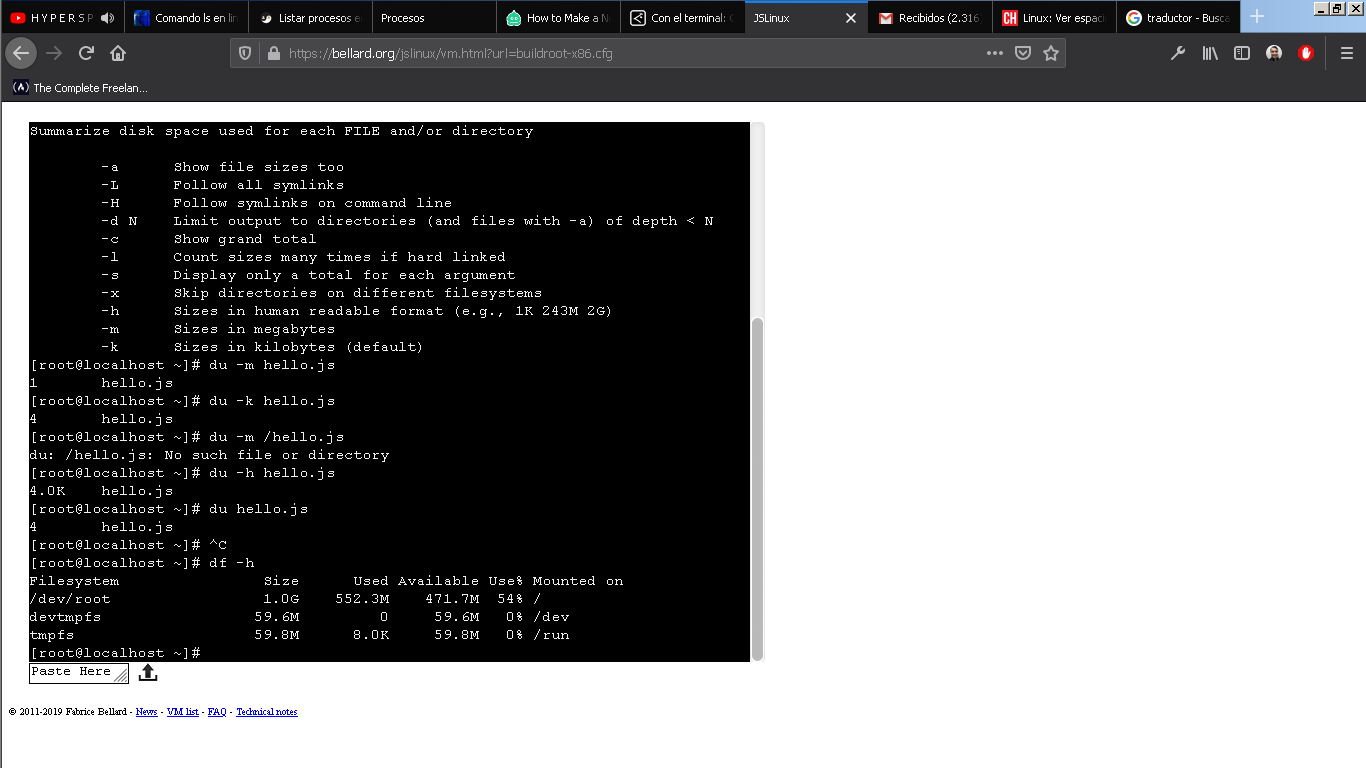


10- ¿Cómo puedo saber cuánto espacio en disco ocupa un directorio completo?

-Con el comando “du” agregando “-k” para saber la medida en kilobytes “-m” en mb o “-h” en formato legible para humanos.

11- ¿Cómo puedo saber cuánto espacio libre hay en un volumen dado?

-Comando “df-h” (<https://computerhoy.com/noticias/software/linux-ver-espacio-libre-disco-78385> )



12- ¿Cómo puedo saber qué otros usuarios están conectados al sistema?

-Comando “w”,”who”, “user”, “last”, ”finger” sin argumentos. (<https://www.sysadmit.com/2018/07/linux-ver-usuarios-conectados.html> )

13- ¿Qué commando ejecutaría (con parámetros) para hacer un backup comprimido del directorio “old” y todos sus subdirectorios?

- tar –zcvf old.tar.gz /home/user/programa

|  |
| --- |
| tar -zcvf nombre-archivo.tar.gz nombre-directorio |

 Donde:

* -z: comprime el archivo usando el programa gzip
* -c: crea el archivo
* -v: verbose. Muestra el progreso mientras se comprime el archivo
* -f: el nombre del archivo a crear

(<https://www.juliocesarpena.com/tutoriales-web/210-como-puedo-comprimir-un-directorio-en-linux-unix> )

14- ¿Cómo haría para recuperar todos los archivos contenidos en un archivo llamado “backup.tgz”

- tar -zxvf backup.tgz “directorio-particular”

-x extrae los archivos