

Solucionario Cuaderno

Tareas Recuperación

Javier Arturo Lucero Ilabaca

26 de octubre del 2023

Indice

Indice	1
Introducción	2
Programación con Python	2
Descripción de algoritmos.	2
Código 1	2
Código 2	3
Código 3	5
Desarrollo de algoritmos	6
Enlace a códigos	6
Enunciado1	6
Enunciado2	7
Enunciado3	7
Enunciado4	7
Enunciado5	7
LINUX	7
Supuesto practico usuarios, grupos y permisos	7
Comandos LINUX	7
Ejercicio 1	7
Ejercicio 2	8
Ejercicio 3	8
Windows	8
Supuestos práctico	8
Gestion de recursos y otros	8
Comandos para la gestion de procesos y memoria	8
Licencia 1	8
Licencia 2	8
Virtualizacion	9
Instalacion dual	9
Red NAT	9

Introducción

Este cuaderno será dividido en 5 puntos principales siendo los bloques a resolver, subiendo con capturas y texto, simultáneamente creando un árbol de directorios en github donde estarán los archivos a todos los ejercicios.

Programación con Python

Nos encontramos con dos apartados.

Descripción de algoritmos.

https://github.com/ArturoLucero28/Lucero_Arturo_ReclSO/tree/66378016a5601117224a8fa69f4a11e4bb9b420b/Python/Descripcion%20de%20algoritmos

Código 1

```
Proceso un_proceso
  definir a, b, c, final Como Entero
  final = 1
  Mientras final ≠ 0 Hacer
    Escribir "escribir tres números distintos, por favor"
    leer a,b,c
    Si a>b Entonces
      Si a>c Entonces
        Escribir " a mayor " a
      Sino
        Escribir "c mayor " c
      Fin Si
    Sino
      Si b>c Entonces
        Escribir "b mayor " b
      Sino
        Escribir "c mayor " c
      Fin Si
    Fin Si
    Escribir "quiere finalizar el proceso teclea 0 ?"
    Leer final
  Fin Mientras
FinProceso
```

Resultado:

#Realiza un script en python que compare 3 números insertados por el usuario, en un bucle donde nos de el mayor de los 3 hasta que el usuario ingrese el 0 y finalice

```
while final != 0:
    print("Escribe tres numeros distintos")
    a = int(input("Ingresa el primer numero: "))
    b = int(input("Ingresa el segundo numero: "))
    c = int(input("Ingresa el tercer numero: "))
    if a > b:
        if a > c:
            print("a es el mayor:", a)
        else:
            print("c es el mayor:", c)
    else:
        if b > c:
            print("b es el mayor:", b)
        else:
            print("c es el mayor:", c)

    print(" Teclee 0 para salir o cualquiera para continuar.")
    final = int(input())

proceso()
```

Código 2

```
Proceso un_proceso
  Definir a,r,c Como Entero
  definir pr Como Caracter
  escribir " desea ejecutar el programa s/n"
  leer pr
  si pr = "s" entonces
    c ← 0
    Repetir
      Leer a
      c ← c+1
      r ← a MOD 2
      Si r=0 Entonces
        Escribir a, ' seleccionado'
      SiNo
        Escribir a, ' no seleccionado'
      FinSi
      Escribir r
    Hasta Que c=10
  FinSi
  escribir "ya terminamos"
FinProceso
```

Resultado:

#Realizar un script en python que pida al usuario si quiere o no realizar el programa y si lo ejecuta pida un numero, con ese numero el programa le dirá si es divisible entre 2, y repetiremos el proceso hasta en 10 ocasiones

```
if pr.lower() == "s":
    c = 0
    while c < 10:
        a = int(input("Ingrese un número: "))
        c += 1
        r = a % 2
        if r == 0:
            print(f"{a} seleccionado")
        else:
            print(f"{a} No seleccionado")
        print(r)

    print("YA TERMINAMOS")
```

Código 3

```
Proceso un_proceso
    definir i, j, r Como Entero
    definir tecla Como Caracter
    i=1
    j=1
    mientras i≠ 0
        escribir "Escribe numero. Pulsar 0 para fin"
        leer i
        si i≠0 entonces
            mientras j<11
                r = i*j
                escribir i " x " j "=" r
                j=j+1
            finmientras
        fin si
        j=1
    fin mientras
FinProceso
```

resultado:

#Realiza un script que de las tablas de multiplicación del numero que ingrese el usuario, tantas veces que quiera el usuario hasta que finalice el programa pulsando el 0

```
def un_proceso():
    i = 1
    j = 1

    while i != 0:
        print("Escribe número o pulsa 0 para finalizar")
        i = int(input())

        if i != 0:
            while j < 11:
                r = i * j
                print(f"{i} x {j} = {r}")
                j = j + 1
            j=1

un_proceso()
```

Desarrollo de algoritmos

Enlace a códigos

https://github.com/ArturoLucero28/Lucero_Arturo_ReclSO/tree/66378016a5601117224a8fa69f4a11e4bb9b420b/Python/Desarrollo%20de%20enunciados

Enunciado 1

Desarrolla un algoritmo que pida al usuario las calificaciones de 5 módulos de un alumno. El algoritmo mostrará en pantalla la calificación mayor, la calificación menor, y la media de las calificaciones.

```
def enunciado1():
    #Creamos una variable en modo lista
    notas = []
    print("Ingresa las calificaciones de los 5 modulos:")
    #un bucle de rango 5 al pedirnos 5 modulos
    for i in range(5):
        modulo = float(input(f"Ingresa una nota: "))
        notas.append(modulo)
        i= i +1
    #Buscamos el mayor,menos y suma de la lista
    mayor = max(notas)
    menor = min(notas)
    suma = sum(notas)
    #Y para la media la conseguimos haciendo la suma de todos diviendolo entre todos los modulos
    media = suma / 5

    print(f"Calificacion mayor: {mayor}")
    print(f"Calificacion menor: {menor}")
    print(f"Media de calificaciones: {media}")

enunciado1()
```

Enunciado 2

Escribe un algoritmo que, dado un número introducido por el usuario, indique si el número es capicua o no.

```
#Crear variables y asignar cadena el numero
numero=input("Introduce un número:")
cadena=str(numero)
#Se invierte
numero_invertido=cadena[::-1]
if cadena == numero_invertido:
    print("El numero es capicúa")
else:
    print("El numero no es capicúa")
```

Enunciado 3

Implementa un algoritmo que busque el nombre de una ciudad entre un conjunto de diez ciudades predefinidas. Al inicio se creará el conjunto con el nombre de las diez ciudades. Posteriormente se pedirá el nombre de una ciudad al usuario. El programa indicará si la ciudad se encuentra en el conjunto o no. Deberá dar por válidas entradas en minúsculas, mayúsculas. Por ejemplo, Si tenemos almacenada “Mérida”, se dará por válido “mérida”, “Mérida”, “MÉRIDA”, “Mérida”, etc.

```
1 #Introducimos las ciudades en una lista
2 ciudades = {"Mérida","Montijo","Guadiana","Caceres","Puebla de la calzada","Lácar","Barbaño","Valdelacalzada","La Roca","Calamonte"}
3 ciudad = input("Introduce el nombre de la ciudad")
4 #para que de igual como lo ingresa el usurio hacemos lower
5 ciudad=ciudad.lower()
6 #verificamos
7 if ciudad in ciudades:
8     print("La ciudad está")
9 else:
10    print("La ciudad no está")
```


Enunciado4

Desarrolla un programa que pida al usuario un número del 1 al 50 y que muestre la representación de dicho número en binario, octal y hexadecimal.

```
1  #pedir numero
2  numero=int(input("Introduce un numero del 1 al 50"))
3  if 0 < numero < 50:
4      #usamos funciones preestablecidas
5      binario= bin(numero)
6      octal= oct(numero)
7      hexadecimal=hex(numero)
8      print(f"Representacion binaria: {binario}")
9      print(f"Representacion octal: {octal}")
10     print(f"Representacion hexadecimal: {hexadecimal}")
11 else:
12     print("Ese numero no se puede")
```



Enunciado5

Implementa un programa que pida al usuario una cantidad de dinero hasta que dicha cantidad sea múltiplo de 5 y, como máximo 3000. Una vez que se cumplan estas condiciones se tiene que mostrar al usuario el número de billetes de 100, 50, 20, 10 y 5 que se necesitan para obtener la cantidad de dinero indicada.



LINUX

Supuesto practico usuarios, grupos y permisos

Comandos LINUX

Ejercicio 1

Indica las instrucciones necesarias para mostrar en la pantalla, en el formato indicado, la siguiente

información de tu usuario:

Nombre de usuario: \$USER

Id de usuario: id -u

Id de grupo: id -g

Ruta de su directorio personal: \$HOME

Ejercicio 2

`sudo useradd [opciones] [nombre]`

usaremos en opciones -c para añadir nombre y apellido, luego el -e YYYY-MM-DD nombre para añadir la fecha de caducidad y con el -m -d es por si la ruta no existe la crearemos.

`sudo useradd -m -d /ruta/ -c "nombre apellido" -e YYYY-MM-DD nombre`

Con el sudo obligamos a que realice la opcion y el useradd añadimos el usuario.

`sudo userdel -r nombredeusuario`

usamos userdel para eliminar el usuario añadiendo el -r [como opción] y el nombre del usuario

Ejercicio 3

- Crear un directorio dentro de Documentos llamado ejercicio_permisos

```
a123@123:~$ mkdir ~/Documentos/ejercicio_permisos
a123@123:~$ cd ~/Documentos/ejercicio_permisos
a123@123:~/Documentos/ejercicio_permisos$
```

- Dentro del directorio crear 9 archivos (archiv1,archiv2,etc.)

```
a123@123:~/Documentos/ejercicio_permisos$ touch archiv{1..9}
a123@123:~/Documentos/ejercicio_permisos$ ls
archiv1  archiv2  archiv3  archiv4  archiv5  archiv6  archiv7  archiv8  archiv9
```

- Establecer a cada archivo los siguientes permisos:

- archiv1 -rwx-----
- archiv2 -rw-----
- archiv3 -rwxrwxrwx
- archiv4 -rwxrw-r--
- archiv5 -rwxr----
- archiv6 -r-xrw-r--
- archiv7 -r-----x
- archiv8 -rw-r--r--
- archiv9 -rw-rw-r--

```

a123@123:~/Documentos/ejercicio_permisos$ chmod 700 archiv1
a123@123:~/Documentos/ejercicio_permisos$ chmod 600 archiv2
a123@123:~/Documentos/ejercicio_permisos$ chmod 777 archiv3
a123@123:~/Documentos/ejercicio_permisos$ chmod 764 archiv4
a123@123:~/Documentos/ejercicio_permisos$ chmod 704 archiv5
a123@123:~/Documentos/ejercicio_permisos$ chmod 614 archiv6
a123@123:~/Documentos/ejercicio_permisos$ chmod 400 archiv7
a123@123:~/Documentos/ejercicio_permisos$ chmod 644 archiv8
a123@123:~/Documentos/ejercicio_permisos$ chmod 664 archiv9
a123@123:~/Documentos/ejercicio_permisos$ ls -l
total 0
-rwx----- 1 a123 a123 0 ene 21 14:47 archiv1
-rw----- 1 a123 a123 0 ene 21 14:47 archiv2
-rwxrwxrwx 1 a123 a123 0 ene 21 14:47 archiv3
-rwxrw-r-- 1 a123 a123 0 ene 21 14:47 archiv4
-rwx---r-- 1 a123 a123 0 ene 21 14:47 archiv5
-rw---xr-- 1 a123 a123 0 ene 21 14:47 archiv6
-r----- 1 a123 a123 0 ene 21 14:47 archiv7
-rw-r--r-- 1 a123 a123 0 ene 21 14:47 archiv8
-rw-rw-r-- 1 a123 a123 0 ene 21 14:47 archiv9

```

- Añade y/o retira permisos sobre los archivos anteriores utilizando los operadores + y - del 'chmod' para que queden de la siguiente manera:

- archiv1 -rwx---r--
- archiv2 -r-----
- archiv3 -rw-rw-rw
- archiv4 -rwx-w----
- archiv5 -rwx----wx
- archiv6 -rwxrw----
- archiv7 -rw---x-w
- archiv8 -----r--
- archiv9 -rwx-----

- Crear un directorio y quitarle todos los permisos de ejecución.

```
a123@123:~/Documentos/ejercicio_permisos$ mkdir directorio_ejemplo
a123@123:~/Documentos/ejercicio_permisos$ chmod 000 directorio_ejemplo
a123@123:~/Documentos/ejercicio_permisos$ cd directorio_ejemplo/
bash: cd: directorio_ejemplo/: Permiso denegado
a123@123:~/Documentos/ejercicio_permisos$
```

- Explicar qué pasa al intentar entrar al directorio con el comando cd.

me da error al no tener el permiso “x” el de ejecución

- Explicar el significado de los permisos r,w y x para directorios.

r : Permite listar el contenido del directorio.

w : Permite crear, eliminar y renombrar archivos en el directorio, así como modificar su contenido.

x : Permite entrar al directorio.

Windows

Supuestos práctico

Gestion de recursos y otros

Comandos para la gestion de procesos y memoria

Licencia 1

Licencia 2



Virtualizacion

Instalacion dual

Red NAT