**ANDERSSON DAVID SÁNCHEZ MÉNDEZ**

**AYPR**

**TAREA 1**

Para los problemas encontrar: Análisis, Diseño y Construcción

1. Escribir un programa que pregunte al usuario su nombre, el número de horas trabajadas y el valor por hora. Después debe mostrar por pantalla el pago que le corresponde.

Análisis

* Contexto: Costo horas trabajadas
* Condiciones iniciales: Se conoce el nombre del usuario, el número de horas trabajadas y el valor por hora trabajada.
* Condiciones finales: Se obtiene el valor del pago por horas trabajadas.

Diseño

* Variables que se necesitan: nombre, horas trabajadas y valor por hora.
* Tipo de datos: nombre(string), horas trabajadas y valor por hora como real
* Algoritmo solución: 1. Preguntar al usuario que diga su nombre.

2. Preguntar al usuario que diga su número de horas trabajadas.

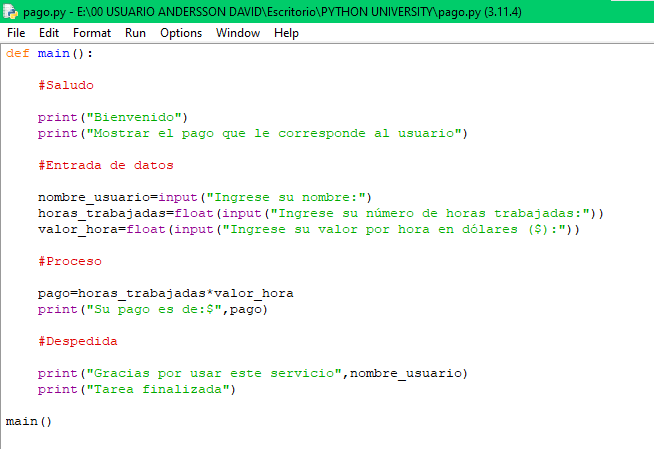
3. Preguntar al usuario que diga su valor por hora trabajada.

4.Realizar la operación necesaria para calcular el pago por horas trabajadas.

5.Decirle a cada usuario el pago por horas trabajadas.

Construcción

* Programa en Python



1. Averiguar si un número entero e pertenece a la unión de los conjuntos formados por los números que pertenecen a los intervalos (a, b] y [5, c)

Análisis

* Contexto: Unión de conjuntos
* Condiciones iniciales: Se conocen los intervalos a la unión de conjuntos.
* Condiciones finales: Se muestra que un número entero llamado e pertenezca a alguno de esos dos intervalos.

Diseño

* Variables que se necesitan: tres números (a, b y c) y un número entero (e).
* Tipo de datos: número e(entero); a, b y c como real.
* Algoritmo solución: 1. Preguntar al usuario que diga 3 números

reales.

2. Preguntar al usuario que diga un número entero.

3. Realizar lo necesario para que el número entero cumpla las condiciones tomando en cuenta los primeros tres números que dijo.

4.Mostrar que el número cumpla o no la condición

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteConstrucción

* Programa en Python

1. Un nivel de azúcar en la sangre menor que 140 mg/dl es normal. Después de horas mayor de 200 mg/dl indica diabetes y entre 140 mg/dl y 199 mg/dl indica prediabetes. Construya un programa en Python que indique el estado de una persona según el nivel de azúcar que reporte.

Análisis

* Contexto: Nivel de azúcar en la sangre
* Condiciones iniciales: nivel de azúcar en la sangre menor a 140 mg/dl es normal, entre 140 y 199 mg/dl indica prediabetes y mayor a 200 indica diabetes.
* Condiciones finales: mostrar el estado de salud de acuerdo con el nivel de azúcar que presente en la sangre

Diseño

* Variables que se necesitan: nivel de azúcar en la sangre (mg/dl) del usuario.
* Tipo de datos: nivel de azúcar en la sangre (float)
* Algoritmo solución: 1. Preguntar al usuario cuál es el nivel de azúcar

en la sangre que tiene.

2. Realizar lo necesario para determinar el estado de salud según el nivel de azúcar en la sangre.

3. Mostrar el estado de salud del usuario.

Construcción

* Programa en Python

Texto, Aplicación

Descripción generada automáticamente

1. Haga un programa en Python que dado un número entero de tres cifras verifique si el dígito de las centenas es igual al de las unidades.

Análisis

* Contexto: Sistema decimal
* Condiciones iniciales: Número entero de tres cifras.
* Condiciones finales: Verificar que el dígito de las centenas sea igual que al de las unidades.

Diseño

* Variables que se necesitan: Número entero de tres cifras.
* Tipo de datos: número de tres cifras (entero)
* Algoritmo solución: 1. Preguntar al usuario que diga un número de

tres cifras.

2.Realizar lo necesario para verificar la condición.

3. Mostrar si el número entero de tres cifras cumple la condición.

Construcción

* Programa en Python

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente