



UNIVERSIDAD
TECNOLÓGICA
DE PANAMÁ

Facultad de Estudios Generales

Asignatura:

DESARROLLO LÓGICO Y ALGORÍTMOS

Estudiante:

Rick Jimenez

4-904-2017

PARCIAL 1

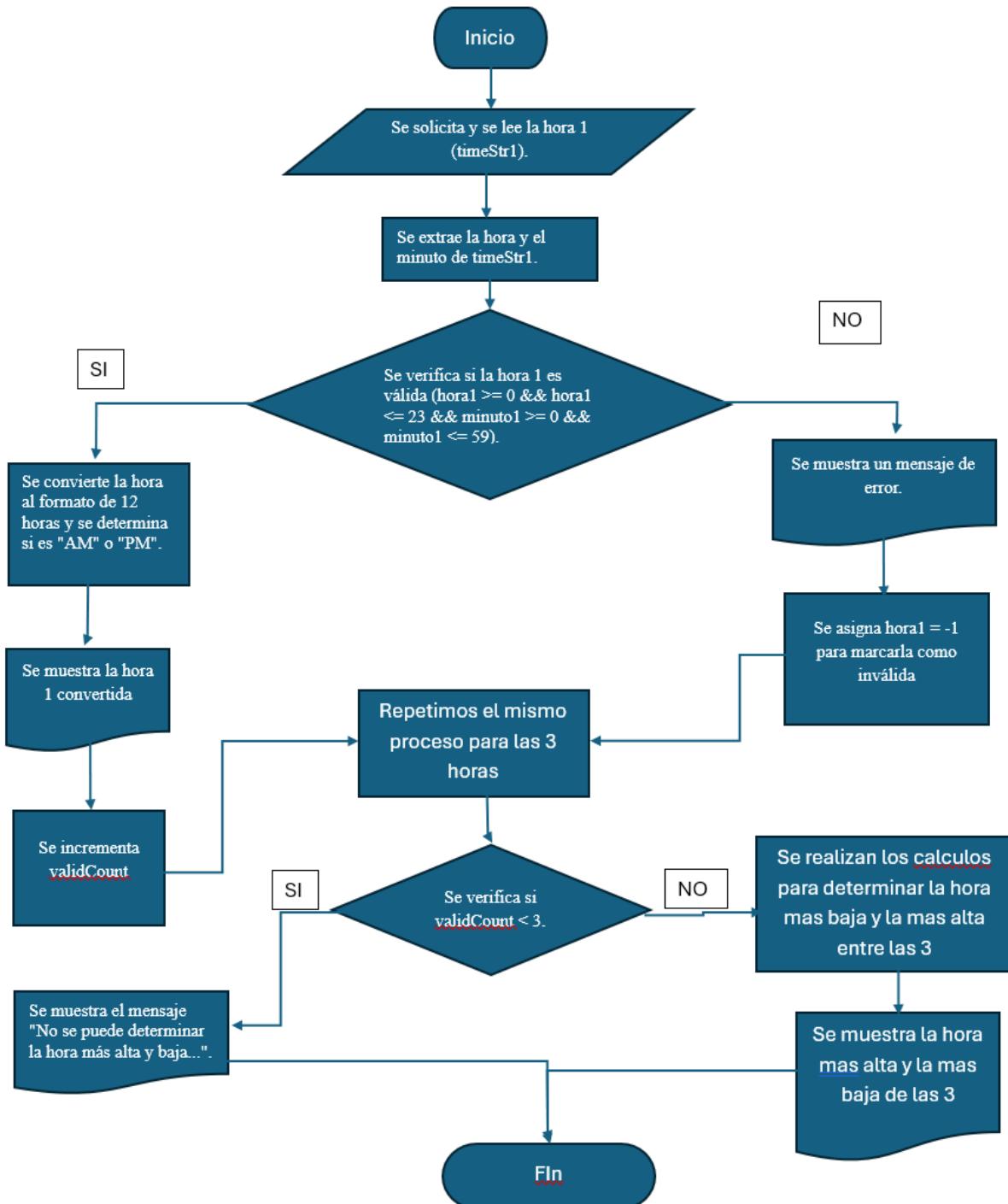
Profesor:

IBARRA, NAPOLEON

Año:

2025

PROBLEMA 1: Escribir una solución que lea la hora (3) en notación de 24 horas y que imprima en notación de 12. Ejemplo: Si la entrada es 13:45, la salida será 1:45 PM. El prototipo de desarrollo debe ser capaz de solicitar al usuario final que introduzca de forma exacta 5 caracteres para especificar 1 hora. Ejemplo: las 9 en punto se debe introducir: 09:00. Esto se debe realizar para 3 valores introducidos por teclado, al final me debe decir también, cuál es la hora más baja y cual es la más alta



PSEUDOCODIGO

```
// Declaración de variables
Definir hora1, minuto1, hora2, minuto2, hora3, minuto3 Como Entero
Definir timeStr1, timeStr2, timeStr3 Como Cadena
Definir horaconvertida1, horaconvertida2, horaconvertida3 Como Entero
Definir ampm1, ampm2, ampm3 Como Cadena
Definir validCount Como Entero
validCount <- 0

// Hora 1
Escribir "Ingrese la hora 1 (HH:MM), 5 caracteres exactos: "
Leer timeStr1
Extraer hora1, minuto1 de timeStr1

Si (hora1 >= 0 y hora1 <= 23 y minuto1 >= 0 y minuto1 <= 59) Entonces
    horaconvertida1 <- hora1
    ampm1 <- "AM"
    Si hora1 >= 12 Entonces
        ampm1 <- "PM"
        Si hora1 > 12 Entonces
            horaconvertida1 <- hora1 - 12
        FinSi
    FinSi
    Si hora1 == 0 Entonces
        horaconvertida1 <- 12
    FinSi
    Escribir "Hora 1 en formato de 12 horas: ", horaconvertida1, ":", minuto1, " ", ampm1
    validCount <- validCount + 1
SiNo
    Escribir "Error: Hora 1 ingresada no es valida."
    hora1 <- -1
FinSi

// Hora 2
Escribir "Ingrese la hora 2 (HH:MM), 5 caracteres exactos: "
Leer timeStr2
Extraer hora2, minuto2 de timeStr2

Si (hora2 >= 0 y hora2 <= 23 y minuto2 >= 0 y minuto2 <= 59) Entonces
    horaconvertida2 <- hora2
    ampm2 <- "AM"
    Si hora2 >= 12 Entonces
        ampm2 <- "PM"
        Si hora2 > 12 Entonces
            horaconvertida2 <- hora2 - 12
        FinSi
    FinSi
    Si hora2 == 0 Entonces
        horaconvertida2 <- 12
    FinSi
```

```

Escribir "Hora 2 en formato de 12 horas: ", horaconvertida2, ":", minuto2, " ", ampm2
validCount <- validCount + 1
SiNo
    Escribir "Error: Hora 2 ingresada no es valida."
    hora2 <- -1
FinSi

// Hora 3
Escribir "Ingrese la hora 3 (HH:MM), 5 caracteres exactos: "
Leer timeStr3
Extraer hora3, minuto3 de timeStr3

Si (hora3 >= 0 y hora3 <= 23 y minuto3 >= 0 y minuto3 <= 59) Entonces
    horaconvertida3 <- hora3
    ampm3 <- "AM"
    Si hora3 >= 12 Entonces
        ampm3 <- "PM"
        Si hora3 > 12 Entonces
            horaconvertida3 <- hora3 - 12
        FinSi
    FinSi
    Si hora3 == 0 Entonces
        horaconvertida3 <- 12
    FinSi
    Escribir "Hora 3 en formato de 12 horas: ", horaconvertida3, ":", minuto3, " ", ampm3
    validCount <- validCount + 1
SiNo
    Escribir "Error: Hora 3 ingresada no es valida."
    hora3 <- -1
FinSi

Escribir ""

Si validCount < 3 Entonces
    Escribir "No se puede determinar la hora más alta y baja porque se ingresaron horas
invalidas."
SiNo
    // Hora más baja
    Si hora1 < hora2 Entonces
        Si hora1 < hora3 Entonces
            Escribir "Hora más baja: ", hora1, ":", minuto1
        SiNo
            Escribir "Hora más baja: ", hora3, ":", minuto3
        FinSi
    SiNo
        Si hora2 < hora3 Entonces
            Escribir "Hora más baja: ", hora2, ":", minuto2
        SiNo
            Escribir "Hora más baja: ", hora3, ":", minuto3
        FinSi
    FinSi

```

```

// Hora más alta
Si hora1 > hora2 Entonces
    Si hora1 > hora3 Entonces
        Escribir "Hora más alta: ", hora1, ":", minuto1
    SiNo
        Escribir "Hora más alta: ", hora3, ":", minuto3
    FinSi
SiNo
    Si hora2 > hora3 Entonces
        Escribir "Hora más alta: ", hora2, ":", minuto2
    SiNo
        Escribir "Hora más alta: ", hora3, ":", minuto3
    FinSi
FinSi
FinSi

```

FinAlgoritmo

CODIGO EN C

```

#include <stdio.h>
#include <string.h>

int main() {
    int hora1, minuto1, hora2, minuto2, hora3, minuto3;
    char timeStr1[6], timeStr2[6], timeStr3[6];
    int horaconvertida1, horaconvertida2, horaconvertida3;
    char ampm1[3], ampm2[3], ampm3[3];
    int validCount = 0; // Contador de horas válidas

    // Hora 1
    printf("Ingrese la hora 1 (HH:MM), 5 caracteres exactos: ");
    scanf("%5s", timeStr1);
    sscanf(timeStr1, "%d:%d", &hora1, &minuto1);

```

```
if (hora1 >= 0 && hora1 <= 23 && minuto1 >= 0 && minuto1 <= 59) {  
    horaconvertida1 = hora1;  
    strcpy(ampm1, "AM");  
    if (hora1 >= 12) {  
        strcpy(ampm1, "PM");  
        if (hora1 > 12) {  
            horaconvertida1 = hora1 - 12;  
        }  
    }  
    if (hora1 == 0) {  
        horaconvertida1 = 12;  
    }  
    printf("Hora 1 en formato de 12 horas: %d:%02d %s\n", horaconvertida1, minuto1,  
ampm1);  
    validCount++;  
} else {  
    printf("Error: Hora 1 ingresada no es valida.\n");  
    hora1 = -1; // Marcar como inválida  
}  
  
// Hora 2  
printf("Ingrese la hora 2 (HH:MM), 5 caracteres exactos: ");  
scanf("%5s", timeStr2);  
sscanf(timeStr2, "%d:%d", &hora2, &minuto2);  
if (hora2 >= 0 && hora2 <= 23 && minuto2 >= 0 && minuto2 <= 59) {  
    horaconvertida2 = hora2;
```

```

strcpy(ampm2, "AM");

if (hora2 >= 12) {
    strcpy(ampm2, "PM");

    if (hora2 > 12) {
        horaconvertida2 = hora2 - 12;
    }
}

if (hora2 == 0) {
    horaconvertida2 = 12;
}

printf("Hora 2 en formato de 12 horas: %d:%02d %s\n", horaconvertida2, minuto2,
ampm2);

validCount++;

} else {
    printf("Error: Hora 2 ingresada no es valida.\n");
    hora2 = -1; // Marcar como inválida
}

// Hora 3

printf("Ingrese la hora 3 (HH:MM), 5 caracteres exactos: ");
scanf("%5s", timeStr3);
sscanf(timeStr3, "%d:%d", &hora3, &minuto3);

if (hora3 >= 0 && hora3 <= 23 && minuto3 >= 0 && minuto3 <= 59) {

    horaconvertida3 = hora3;

    strcpy(ampm3, "AM");

    if (hora3 >= 12) {

```

```

strcpy(ampm3, "PM");

if (hora3 > 12) {
    horaconvertida3 = hora3 - 12;
}

}

if (hora3 == 0) {
    horaconvertida3 = 12;
}

printf("Hora 3 en formato de 12 horas: %d:%02d %s\n", horaconvertida3, minuto3,
ampm3);

validCount++;

} else {

    printf("Error: Hora 3 ingresada no es valida.\n");
    hora3 = -1; // Marcar como inválida
}

printf("\n");

// Encontrar la hora más alta y la más baja

if (validCount < 3) {

    printf("No se puede determinar la hora más alta y baja porque se ingresaron horas
invalidas.\n");

} else {

    // Hora más baja

    if (hora1 < hora2) {

        if (hora1 < hora3) {

            printf("Hora más baja: %d:%02d\n", hora1, minuto1);
}

```

```
    } else {
        printf("Hora más baja: %d:%02d\n", hora3, minuto3);
    }

} else {
    if (hora2 < hora3) {
        printf("Hora más baja: %d:%02d\n", hora2, minuto2);
    } else {
        printf("Hora más baja: %d:%02d\n", hora3, minuto3);
    }
}

// Hora más alta

if (hora1 > hora2) {
    if (hora1 > hora3) {
        printf("Hora más alta: %d:%02d\n", hora1, minuto1);
    } else {
        printf("Hora más alta: %d:%02d\n", hora3, minuto3);
    }
} else {
    if (hora2 > hora3) {
        printf("Hora más alta: %d:%02d\n", hora2, minuto2);
    } else {
        printf("Hora más alta: %d:%02d\n", hora3, minuto3);
    }
}
}
```

```
    return 0;  
}
```

PROBLEMA 2:

Crear una solución que acepte 3 fechas escritas de forma usual (como "15, febrero, 1989") y las visualice en 3 números (como "15 2 1989"). El programa debe determinar y mostrar cuál es la fecha más baja y cuál es la fecha más alta de las tres.

PSEUDOCODIGO

```
// Definición de variables  
Definir day1, year1, month1 Como Entero  
Definir monthStr1 Como Cadena  
Definir day2, year2, month2 Como Entero  
Definir monthStr2 Como Cadena  
Definir day3, year3, month3 Como Entero  
Definir monthStr3 Como Cadena  
Definir lowestDay, lowestMonth, lowestYear Como Entero  
Definir highestDay, highestMonth, highestYear Como Entero  
  
// Obtener y procesar la primera fecha  
Escribir "Ingrese la fecha 1 (ej. 15, febrero, 1989): "  
Leer day1, monthStr1, year1  
  
// Convertir el nombre del mes a número  
Si monthStr1 == "enero," Entonces month1 <- 1 FinSi  
Si monthStr1 == "febrero," Entonces month1 <- 2 FinSi  
Si monthStr1 == "marzo," Entonces month1 <- 3 FinSi  
Si monthStr1 == "abril," Entonces month1 <- 4 FinSi  
Si monthStr1 == "mayo," Entonces month1 <- 5 FinSi  
Si monthStr1 == "junio," Entonces month1 <- 6 FinSi  
Si monthStr1 == "julio," Entonces month1 <- 7 FinSi  
Si monthStr1 == "agosto," Entonces month1 <- 8 FinSi  
Si monthStr1 == "septiembre," Entonces month1 <- 9 FinSi  
Si monthStr1 == "octubre," Entonces month1 <- 10 FinSi  
Si monthStr1 == "noviembre," Entonces month1 <- 11 FinSi
```

```

Si monthStr1 == "diciembre," Entonces month1 <- 12 FinSi
Escribir "Fecha 1 en formato numerico:", day1, " ", month1, " ", year1

// Inicializar los valores de la fecha más alta y más baja
lowestDay <- day1
lowestMonth <- month1
lowestYear <- year1
highestDay <- day1
highestMonth <- month1
highestYear <- year1

// Obtener y procesar la segunda fecha
Escribir "Ingrese la fecha 2 (ej. 15, febrero, 1989): "
Leer day2, monthStr2, year2
// Convertir el nombre del mes a número
Si monthStr2 == "enero," Entonces month2 <- 1 FinSi
...
Escribir "Fecha 2 en formato numerico:", day2, " ", month2, " ", year2

// Comparar con la fecha más alta y más baja actual
Si (year2 < lowestYear) o (year2 == lowestYear y month2 < lowestMonth) o (year2 ==
lowestYear y month2 == lowestMonth y day2 < lowestDay) Entonces
    lowestDay <- day2
    lowestMonth <- month2
    lowestYear <- year2
FinSi
Si (year2 > highestYear) o (year2 == highestYear y month2 > highestMonth) o (year2 ==
highestYear y month2 == highestMonth y day2 > highestDay) Entonces
    highestDay <- day2
    highestMonth <- month2
    highestYear <- year2
FinSi

// Obtener y procesar la tercera fecha
Escribir "Ingrese la fecha 3 (ej. 15, febrero, 1989): "
Leer day3, monthStr3, year3
// Convertir el nombre del mes a número
Si monthStr3 == "enero," Entonces month3 <- 1 FinSi

```

...

Escribir "Fecha 3 en formato numerico: ", day3, " ", month3, " ", year3

// Comparar con la fecha más alta y más baja actual

Si (year3 < lowestYear) o (year3 == lowestYear y month3 < lowestMonth) o (year3 == lowestYear y month3 == lowestMonth y day3 < lowestDay) Entonces

lowestDay <- day3

lowestMonth <- month3

lowestYear <- year3

FinSi

Si (year3 > highestYear) o (year3 == highestYear y month3 > highestMonth) o (year3 == highestYear y month3 == highestMonth y day3 > highestDay) Entonces

highestDay <- day3

highestMonth <- month3

highestYear <- year3

FinSi

Escribir ""

Escribir "La fecha más baja es: ", lowestDay, " ", lowestMonth, " ", lowestYear

Escribir "La fecha más alta es: ", highestDay, " ", highestMonth, " ", highestYear

FinAlgoritmo

CODIGO EN C

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int main() {
```

```
    int day1, year1, month1;
```

```
    char monthStr1[20];
```

```
    char comma;
```

```
    int day2, year2, month2;
```

```
    char monthStr2[20];
```

```
int day3, year3, month3;  
  
char monthStr3[20];  
  
  
// Variables para almacenar la fecha más baja y más alta  
  
int lowestDay, lowestMonth, lowestYear;  
  
int highestDay, highestMonth, highestYear;  
  
  
// Obtener y procesar la primera fecha  
  
printf("Ingrese la fecha 1 (ej. 15, febrero, 1989): ");  
  
scanf("%d%c %s %d", &day1, &comma, monthStr1, &year1);  
  
  
// Convertir el nombre del mes a número para la primera fecha  
  
if (strcmp(monthStr1, "enero,") == 0) month1 = 1;  
  
if (strcmp(monthStr1, "febrero,") == 0) month1 = 2;  
  
if (strcmp(monthStr1, "marzo,") == 0) month1 = 3;  
  
if (strcmp(monthStr1, "abril,") == 0) month1 = 4;  
  
if (strcmp(monthStr1, "mayo,") == 0) month1 = 5;  
  
if (strcmp(monthStr1, "junio,") == 0) month1 = 6;  
  
if (strcmp(monthStr1, "julio,") == 0) month1 = 7;  
  
if (strcmp(monthStr1, "agosto,") == 0) month1 = 8;  
  
if (strcmp(monthStr1, "septiembre,") == 0) month1 = 9;  
  
if (strcmp(monthStr1, "octubre,") == 0) month1 = 10;  
  
if (strcmp(monthStr1, "noviembre,") == 0) month1 = 11;  
  
if (strcmp(monthStr1, "diciembre,") == 0) month1 = 12;  
  
printf("Fecha 1 en formato numerico: %d %d %d\n", day1, month1, year1);
```

```
// Inicializar las variables de fecha más alta y más baja  
  
lowestDay = day1;  
  
lowestMonth = month1;  
  
lowestYear = year1;  
  
highestDay = day1;  
  
highestMonth = month1;  
  
highestYear = year1;  
  
  
// Obtener y procesar la segunda fecha  
  
printf("Ingrese la fecha 2 (ej. 15, febrero, 1989): ");  
  
scanf("%d%c %s %d", &day2, &comma, monthStr2, &year2);  
  
  
// Convertir el nombre del mes a número para la segunda fecha  
  
if (strcmp(monthStr2, "enero,") == 0) month2 = 1;  
  
if (strcmp(monthStr2, "febrero,") == 0) month2 = 2;  
  
if (strcmp(monthStr2, "marzo,") == 0) month2 = 3;  
  
if (strcmp(monthStr2, "abril,") == 0) month2 = 4;  
  
if (strcmp(monthStr2, "mayo,") == 0) month2 = 5;  
  
if (strcmp(monthStr2, "junio,") == 0) month2 = 6;  
  
if (strcmp(monthStr2, "julio,") == 0) month2 = 7;  
  
if (strcmp(monthStr2, "agosto,") == 0) month2 = 8;  
  
if (strcmp(monthStr2, "septiembre,") == 0) month2 = 9;  
  
if (strcmp(monthStr2, "octubre,") == 0) month2 = 10;  
  
if (strcmp(monthStr2, "noviembre,") == 0) month2 = 11;  
  
if (strcmp(monthStr2, "diciembre,") == 0) month2 = 12;
```

```

printf("Fecha 2 en formato numerico: %d %d %d\n", day2, month2, year2);

// Comparar con la fecha más alta y más baja actual

if (year2 < lowestYear || (year2 == lowestYear && month2 < lowestMonth) || (year2 ==
lowestYear && month2 == lowestMonth && day2 < lowestDay)) {

    lowestDay = day2;

    lowestMonth = month2;

    lowestYear = year2;

}

if (year2 > highestYear || (year2 == highestYear && month2 > highestMonth) || (year2 ==
highestYear && month2 == highestMonth && day2 > highestDay)) {

    highestDay = day2;

    highestMonth = month2;

    highestYear = year2;

}

// Obtener y procesar la tercera fecha

printf("Ingrese la fecha 3 (ej. 15, febrero, 1989): ");

scanf("%d%c %s %d", &day3, &comma, monthStr3, &year3);

// Convertir el nombre del mes a número para la tercera fecha

if (strcmp(monthStr3, "enero,") == 0) month3 = 1;

if (strcmp(monthStr3, "febrero,") == 0) month3 = 2;

if (strcmp(monthStr3, "marzo,") == 0) month3 = 3;

if (strcmp(monthStr3, "abril,") == 0) month3 = 4;

if (strcmp(monthStr3, "mayo,") == 0) month3 = 5;

if (strcmp(monthStr3, "junio,") == 0) month3 = 6;

```

```

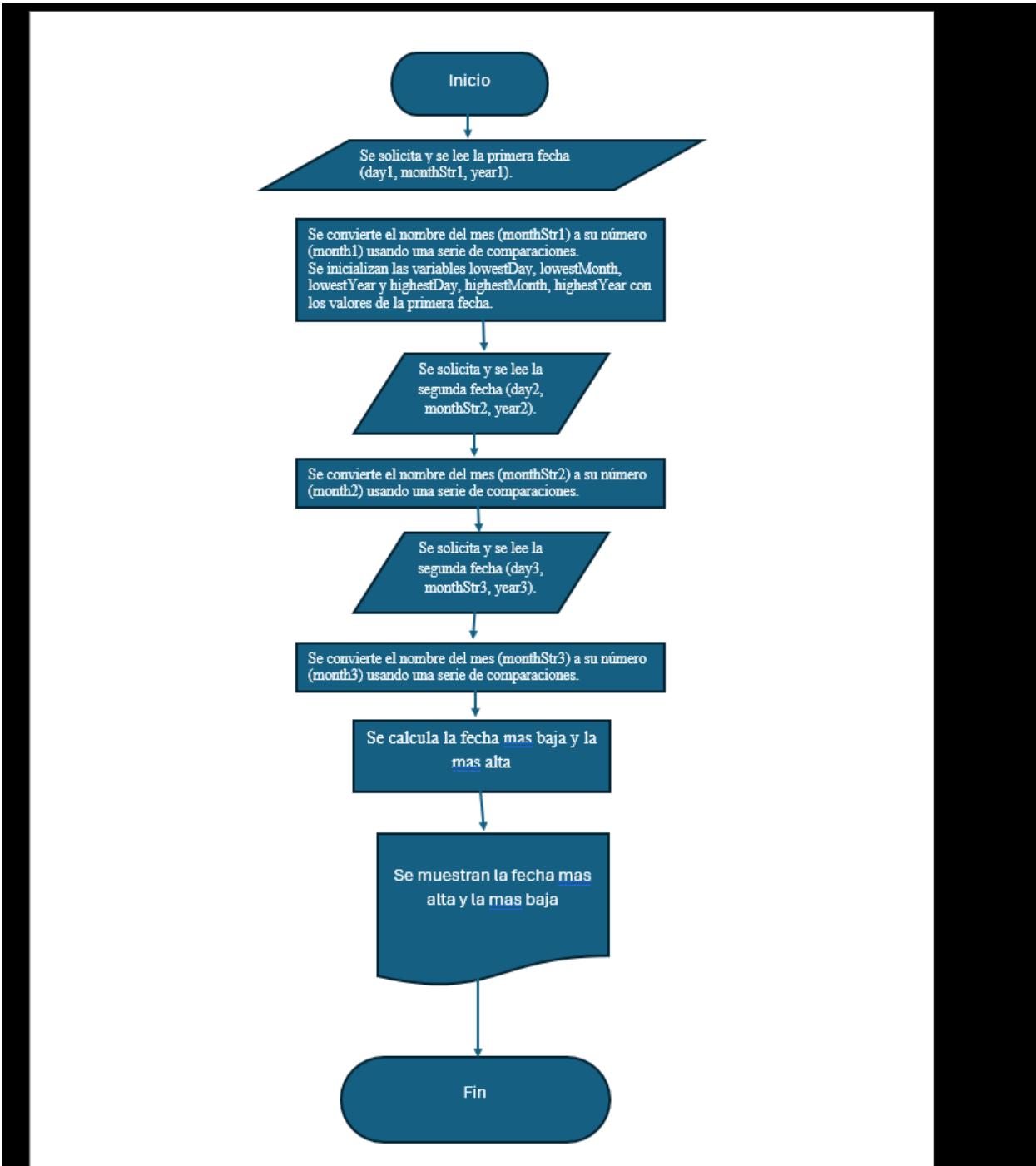
if (strcmp(monthStr3, "julio,") == 0) month3 = 7;
if (strcmp(monthStr3, "agosto,") == 0) month3 = 8;
if (strcmp(monthStr3, "septiembre,") == 0) month3 = 9;
if (strcmp(monthStr3, "octubre,") == 0) month3 = 10;
if (strcmp(monthStr3, "noviembre,") == 0) month3 = 11;
if (strcmp(monthStr3, "diciembre,") == 0) month3 = 12;
printf("Fecha 3 en formato numerico: %d %d %d\n", day3, month3, year3);

// Comparar con la fecha más alta y más baja actual
if (year3 < lowestYear || (year3 == lowestYear && month3 < lowestMonth) || (year3 == lowestYear && month3 == lowestMonth && day3 < lowestDay)) {
    lowestDay = day3;
    lowestMonth = month3;
    lowestYear = year3;
}

if (year3 > highestYear || (year3 == highestYear && month3 > highestMonth) || (year3 == highestYear && month3 == highestMonth && day3 > highestDay)) {
    highestDay = day3;
    highestMonth = month3;
    highestYear = year3;
}

printf("\n");
printf("La fecha más baja es: %d %d %d\n", lowestDay, lowestMonth, lowestYear);
printf("La fecha más alta es: %d %d %d\n", highestDay, highestMonth, highestYear);
return 0;
}

```



PROBLEMA 3:

Dadas dos fechas, la de nacimiento y la actual, en formato día, mes y año. La solución debe calcular y mostrar la edad del individuo. Si la persona tiene menos de un año, la edad se debe mostrar en meses y días ; de lo contrario, debe mostrarse en años.

PSEUDOCODIGO

Algoritmo CalcularEdad

// Definición de variables

Definir birthDay, birthMonth, birthYear Como Entero

Definir currentDay, currentMonth, currentYear Como Entero

Definir ageYears, ageMonths, ageDays Como Entero

Escribir "Ingrese la fecha de nacimiento (dia mes anio): "

Leer birthDay, birthMonth, birthYear

Escribir "Ingrese la fecha actual (dia mes anio): "

Leer currentDay, currentMonth, currentYear

ageYears <- currentYear - birthYear

ageMonths <- currentMonth - birthMonth

ageDays <- currentDay - birthDay

// Ajustar los meses y días si es necesario

Si ageDays < 0 Entonces

 ageMonths <- ageMonths - 1

 ageDays <- ageDays + 30

FinSi

Si ageMonths < 0 Entonces

 ageYears <- ageYears - 1

 ageMonths <- ageMonths + 12

FinSi

Escribir ""

// Mostrar la edad según la condición de ser menor de un año

Si ageYears == 0 Entonces

 Escribir "Edad: ", ageMonths, " meses y ", ageDays, " dias"

SiNo

 Escribir "Edad: ", ageYears, " anios"

FinSi

FinAlgoritmo

CODIGO EN C

```
#include <stdio.h>

int main() {
    int birthDay, birthMonth, birthYear;
    int currentDay, currentMonth, currentYear;
    int ageYears, ageMonths, ageDays;

    printf("Ingrese la fecha de nacimiento (dia mes año): ");
    scanf("%d %d %d", &birthDay, &birthMonth, &birthYear);

    printf("Ingrese la fecha actual (dia mes año): ");
    scanf("%d %d %d", &currentDay, &currentMonth, &currentYear);

    ageYears = currentYear - birthYear;
    ageMonths = currentMonth - birthMonth;
    ageDays = currentDay - birthDay;

    // Ajustar los meses y días si la fecha actual es anterior a la de nacimiento en el
    // mismo año.

    if (ageDays < 0) {
        ageMonths = ageMonths - 1;
        ageDays = ageDays + 30;
    }

    if (ageMonths < 0) {
        ageYears = ageYears - 1;
        ageMonths = ageMonths + 12;
    }
}
```

```
}

printf("\n");

// Imprimir la edad según la condición de si es menor de un año.

if (ageYears == 0) {

    printf("Edad: %d meses y %d días\n", ageMonths, ageDays);

} else {

    printf("Edad: %d años\n", ageYears);

}

return 0;

}
```



PROBLEMA 4:

Codificar una solución que determine si un año es bisiesto. Un año es bisiesto si es múltiplo de 4, excepto si también es múltiplo de 100. Los años múltiples de 100 solo son bisiestos si también son múltiplos de 400. Por ejemplo, 1800 no es bisiesto, pero 2000 sí lo es.

PSEUDOCODIGO

Fragmento de código

```
Algoritmo AnioBisiesto
    // Definición de variables
    Definir year Como Entero

    Escribir "Ingrese un año para verificar si es bisiesto: "
    Leer year

    // Determinar si es bisiesto
    Si (year % 4 == 0) Entonces
        Si (year % 100 == 0) Entonces
            Si (year % 400 == 0) Entonces
                Escribir year, " es un año bisiesto."
            FinSi
        SiNo
            Escribir year, " no es un año bisiesto."
        FinSi
    SiNo
        Escribir year, " es un año bisiesto."
    FinSi
FinAlgoritmo
```

CODIGO EN C

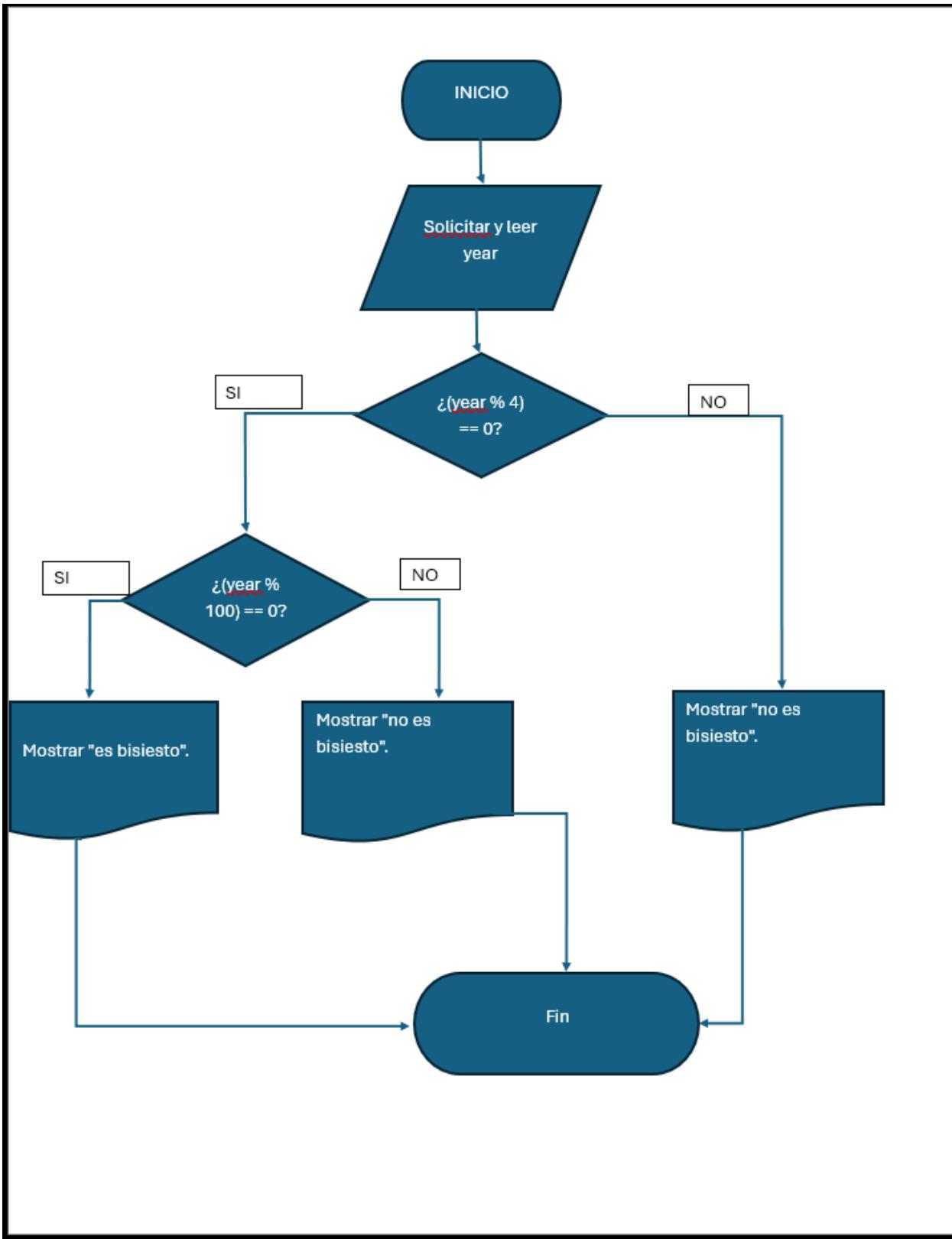
```
#include <stdio.h>
```

```
int main() {
    int year;

    printf("Ingrese un año para verificar si es bisiesto: ");
    scanf("%d", &year);

    if (year % 4 == 0) {
        if (year % 100 == 0) {
            if (year % 400 == 0) {
                printf("%d es un año bisiesto.\n", year);
            } else {
                printf("%d no es un año bisiesto.\n", year);
            }
        } else {
            printf("%d es un año bisiesto.\n", year);
        }
    } else {
        printf("%d no es un año bisiesto.\n", year);
    }

    return 0;
}
```



PROBLEMA 5:

Crear una solución que calcule el salario neto semanal de un trabajador. Las horas trabajadas hasta 40 son a una tasa normal. Las horas extras (41 o más) se pagan con una tasa 50% superior a la ordinaria. El impuesto es del 0% si el salario bruto es menor o igual a 750 Balboas, y del 10% si es mayor.

PSEUDOCODIGO:

Algoritmo SalarioNeto

// Definición de variables

Definir hoursWorked, hourlyRate, grossSalary, netSalary Como Real

Definir taxRate Como Real

taxRate <- 0.0

Escribir "Ingrese las horas semanales trabajadas: "

Leer hoursWorked

Escribir "Ingrese la tarifa por hora: "

Leer hourlyRate

// Calcular el salario bruto

Si hoursWorked <= 40 Entonces

 grossSalary <- hoursWorked * hourlyRate

SiNo

 grossSalary <- (40 * hourlyRate) + ((hoursWorked - 40) * (hourlyRate * 1.5))

FinSi

// Determinar la tasa de impuestos

Si grossSalary > 750 Entonces

 taxRate <- 0.10

FinSi

// Calcular el salario neto

netSalary <- grossSalary - (grossSalary * taxRate)

Escribir ""

Escribir "Salario Bruto: ", Formato(grossSalary, 2), " Balboas"

Escribir "Salario Neto: ", Formato(netSalary, 2), " Balboas"

FinAlgoritmo

CODIGO EN C

```
#include <stdio.h>

int main() {
    float hoursWorked, hourlyRate, grossSalary, netSalary;
    float taxRate = 0.0;

    printf("Ingrese las horas semanales trabajadas: ");
    scanf("%f", &hoursWorked);

    printf("Ingrese la tarifa por hora: ");
    scanf("%f", &hourlyRate);

    // Calcular el salario bruto
    if (hoursWorked <= 40) {
        grossSalary = hoursWorked * hourlyRate;
    } else {
        grossSalary = (40 * hourlyRate) + ((hoursWorked - 40) * (hourlyRate * 1.5));
    }

    // Determinar la tasa de impuestos
    if (grossSalary > 750) {
        taxRate = 0.10;
    }
}
```

```
// Calcular el salario neto  
  
netSalary = grossSalary - (grossSalary * taxRate);  
  
  
printf("\n");  
  
printf("Salario Bruto: %.2f Balboas\n", grossSalary);  
  
printf("Salario Neto: %.2f Balboas\n", netSalary);  
  
  
return 0;  
}
```

