

28 Preguntas del 5.4d:

¿Qué debe mostrar como resultado?

El programa debe mostrar como resultado el signo zodiacal correspondiente a la fecha de nacimiento que el usuario ingresó. Por ejemplo:

Su signo zodiacal es: Leo

En caso de que el usuario ingrese una fecha inválida (por ejemplo, 31 de abril o 30 de febrero), el programa debe mostrar un mensaje como:

Fecha inválida.

Esto ayuda a que el usuario sepa si ingresó correctamente su información o si necesita corregir algún dato.

```
1 #include <iostream>
2
3
4 std::string signoZodiacal(int dia, int mes) {
5     if ((mes == 3 && dia >= 21) || (mes == 4 && dia <= 19)) return "Aries";
6     if ((mes == 4 && dia >= 20) || (mes == 5 && dia <= 20)) return "Tauro";
7     if ((mes == 5 && dia >= 21) || (mes == 6 && dia <= 20)) return "Géminis";
8     if ((mes == 6 && dia >= 21) || (mes == 7 && dia <= 22)) return "Cáncer";
9     if ((mes == 7 && dia >= 23) || (mes == 8 && dia <= 22)) return "Leo";
10    if ((mes == 8 && dia >= 23) || (mes == 9 && dia <= 22)) return "Virgo";
11    if ((mes == 9 && dia >= 23) || (mes == 10 && dia <= 22)) return "Libra";
12    if ((mes == 10 && dia >= 23) || (mes == 11 && dia <= 21)) return "Escorpio";
13    if ((mes == 11 && dia >= 22) || (mes == 12 && dia <= 21)) return "Sagitario";
14    if ((mes == 12 && dia >= 22) || (mes == 1 && dia <= 19)) return "Capricornio";
15    if ((mes == 1 && dia >= 20) || (mes == 2 && dia <= 18)) return "Acuario";
16    if ((mes == 2 && dia >= 19) || (mes == 3 && dia <= 20)) return "Piscis";
17    return "Fecha inválida";
18 }
19
20 int main() {
21     int dia, mes;
22     std::cout << "Ingrese día de nacimiento: ";
23     std::cin >> dia;
24     std::cout << "Ingrese mes de nacimiento (1-12): ";
25     std::cin >> mes;
26
27     std::cout << "Su signo zodiacal es: " << signoZodiacal(dia, mes) << std::endl;
28     return 0;
29 }
```

Figure 10: código

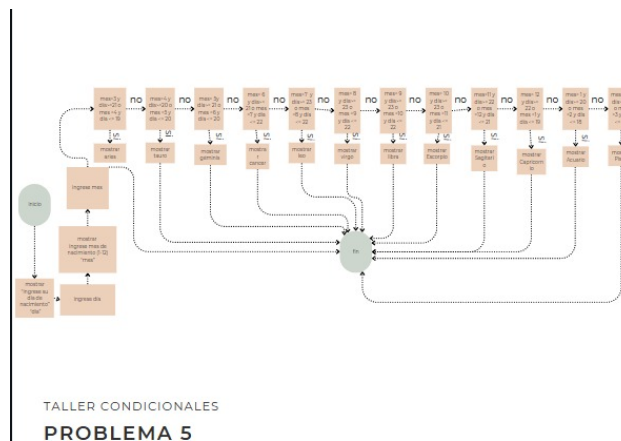


Figure 11: código

29 Preguntas del 6.f:

¿Qué tipo de programa necesitas crear?

Necesitas crear un programa en C++ que permita al usuario ingresar una hora en formato militar (24 horas) y un minuto, y luego verifique si ambos valores son válidos. Si lo son, muestra el horario ingresado; si no, muestra un mensaje de error.

30 Preguntas del 6.1f:

¿Qué validaciones son necesarias?

El programa realiza dos validaciones:

Validación de la hora (x):

Debe estar entre 0 y 23 (aunque tu código acepta hasta 23.999... debido a `x < 24`).

Validación de los minutos (name):

Deben estar entre 0 y 59.

Ambas validaciones están correctamente hechas con condicionales `if`.

31 Preguntas del 6.2f:

¿Qué casos especiales debes considerar?

Algunos casos especiales que podrías considerar para mejorar el programa:

Si el usuario ingresa letras en lugar de números (esto causaría un error en `std::cin`).

Si se ingresan valores fuera de rango, como:

Hora: -1, 24, 100, etc.

Minutos: -5, 60, etc.

Considerar agregar un formato más legible si el usuario ingresa por ejemplo 7:5, que se muestre como 07:05.

32 Preguntas del 6.3f:

¿Qué entrada necesita el programa del usuario?

El programa necesita dos entradas por parte del usuario:

Un número entero para la hora en formato militar (entre 0 y 23).

Un número entero para los minutos (entre 0 y 59).

33 Preguntas del 6.4f:

¿Qué debe mostrar como resultado?

El horario militar da el resultado que el horario militar pueda permitir el ingreso y que pueda cancelar y poner el ingreso después de cada hora como se puede ver en las migas vistas

gracias horario aceptado horario ingresado fue hh:mm

gracias horario aceptado horario ingresado fue 14:30

hora invalida verifique y marque nuevamente

```

1 #include <iostream>
2 int main () {
3     int name;
4     int x;
5
6     std::cout << "ingrese horario en hora militar" << std::endl;
7     std::cin >> x;
8
9     if (x < 24 && x > -1) {
10         std::cout << "ingrese minuto de horario" << std::endl;
11         std::cin >> name;
12         if (name < 60 && name > -1) {
13             std::cout << "gracias horario aceptado" << std::endl;
14             std::cout << "horario ingresado fue" << std::endl;
15             std::cout << x << ":" << name << std::endl;
16         } else {
17             std::cout << "hora invalida" << std::endl;
18             std::cout << "verifique y marque nuevamente" << std::endl;
19         }
20     } else {
21         std::cout << "hora invalida" << std::endl;
22         std::cout << "verifique y marque nuevamente" << std::endl;
23     }
24 }
25
26
27

```

Figure 12: codigo

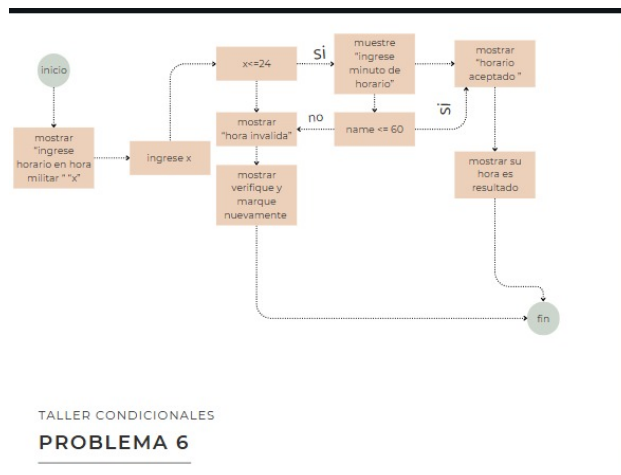


Figure 13: codigo