

Fundamentos de la programación y la informática
Examen Parcial. 7 de Noviembre de 2024
Grados en ingeniería aeroespacial. Turno de tarde
Universidad Rey Juan Carlos

Ejercicio 1 (2 puntos)

Contesta en el fichero `~/logica.txt`. Sean las expresiones

```
e1 := not (not a or not b) or (not c and not d) or (e and not f);
```

```
e2 := not (x > 4) or (y <> 2);
```

1. A partir de `e1`, aplica De Morgan y escribe una expresión lógica equivalente, en Pascal, intentando que sea más clara, de nombre `s1`
2. A partir de `e2`, escribe una expresión lógica equivalente, en Pascal, más legible, de nombre `s2`
3. Escribe una expresión `n1`, de la forma más clara posible, que sea la negación de `s1`
4. Escribe una expresión `n2`, de la forma más clara posible, que sea la negación de `s2`

Solución

```
s1 := (a and b) or (not c and not d) or (e and not f)
```

También podemos expresarla de esta otra forma, de claridad muy similar

```
s1 := (a and b) or not (c or d) or (e and not f)
```

```
s2 := (x <= 4) or (y <> 2)
```

```
n1 := (not a or not b) and (c or d) and (not e or f)
```

```
n2 := (x > 4) and (y = 2)
```

Ejercicio 2 (8 puntos)

`~/circuito.pas` contiene el esqueleto de un programa en Pascal.

También está aquí: <https://gsyc.urjc.es/~mortuno/fpi/circuito.pas>. Complétalo para para cumplir la siguiente especificación:

Deseamos dar indicaciones para un ejercicio de navegación en el que una aeronave debe seguir un patrón de tráfico estándar en torno a una pista de despegue con una orientación específica. El circuito tiene forma rectangular, cada lado se denomina *tramo*, está formado por los tramos *viento en cara*, *viento cruzado*, *viento en cola*, *base*, *final*. Según el siguiente esquema (figura 1):

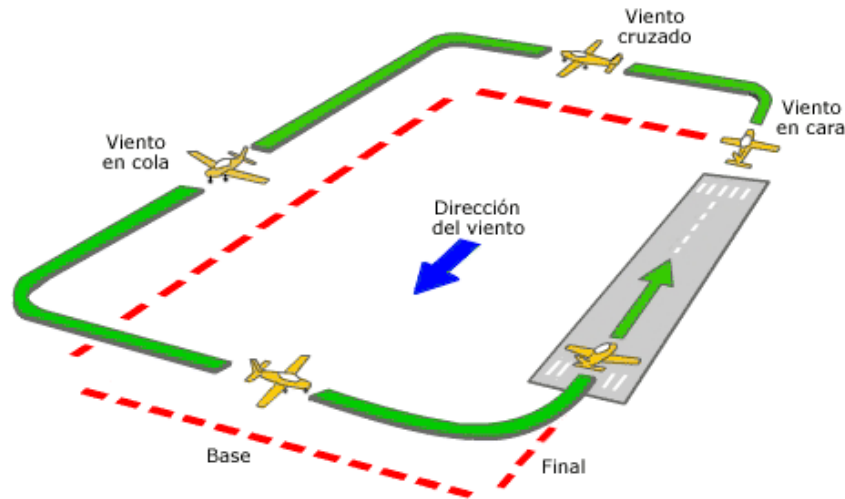


Figura 1: Circuito de aeródromo

Fuente: *manualvuelo.es*

El programa debe calcular los *headings* (figura 2, direcciones en grados respecto al norte magnético) para cada uno de los tramos del circuito, también el tiempo necesario para recorrer los tramos segundo, tercero y cuarto. Y con todo esto, generar las instrucciones de navegación correspondientes.

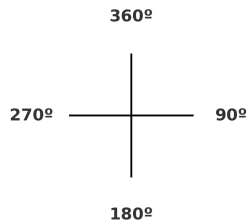


Figura 2: Heading

Valores de entrada:

1. Número de pista. Será un número entero entre 1 y 36. Ambos inclusive.
A partir del número de pista conocemos su orientación: basta multiplicar por 10. Por ejemplo, la pista 10 tiene una orientación de 100° .
2. Velocidad de la aeronave, en nudos. Por simplificar la supondremos constante. Será un número entero entre 90 y 150, ambos inclusive.
3. Distancia a recorrer en el tramo de viento en cola (el tramo *largo*). Expresada en millas náuticas. Será un número real entre 0.5 y 5, ambos inclusive.

A partir de esta, podremos conocer la distancia a recorrer en los tramos de viento cruzado y base: la mitad. Por ejemplo, si el tramo viento en cola son 4 millas, los tramos viento cruzado y tramo base (los tramos *cortos*) serán de 2 millas cada uno.

Valores de salida:

1. Para cada cambio de dirección en el circuito, el *heading* varía 90° hacia la izquierda (en sentido antihorario). El programa debe asegurarse de que los *headings* resultantes estén dentro del rango de 1° a 360° , sumando 360° si el resultado es negativo.

En otras palabras, se trata simplemente de restar, tres veces, 90 al valor inicial. Sumando 360 al resultado si el valor es cero o negativo.

2. Tiempo necesario para recorrer el tramo de viento en cola. Se calculará como distancia del tramo *largo* entre la velocidad, multiplicando por 3600 para expresarlo en segundos, no en horas. $t = 3600 * \frac{\text{distancia}}{\text{velocidad}}$

Aquí tienes un ejemplo de la salida del programa:

```
Número de pista: 18 Velocidad: 120 nudos Distancia: 4.0 millas
Tramo inicial: 180°.
Tramo viento cruzado: 90°. 60 segundos
Tramo viento en cola: 360°. 120 segundos
Tramo base: 270°. 60 segundos
Tramo viento de frente: 180°.
```

```
Número de pista: 10 Velocidad: 100 nudos Distancia: 2.0 millas
Tramo inicial: 100°.
Tramo viento cruzado: 10°. 36 segundos
Tramo viento en cola: 280°. 72 segundos
Tramo base: 190°. 36 segundos
Tramo viento de frente: 100°.
```

```
Número de pista: 40 Velocidad: 100 nudos Distancia: 2.0 millas
Número de pista fuera de rango
```

```
Número de pista: 18 Velocidad: 89 nudos Distancia: 2.0 millas
Velocidad fuera de rango
```

```
Número de pista: 18 Velocidad: 151 nudos Distancia: 2.0 millas
Velocidad fuera de rango
```

```
Número de pista: 18 Velocidad: 110 nudos Distancia: 0.4 millas
Distancia fuera de rango
```

```
Número de pista: 18 Velocidad: 110 nudos Distancia: 5.5 millas
Distancia fuera de rango
```

Instrucciones adicionales

1. Usa todas las variables y subprogramas del esqueleto. No borres nada.
2. El programa trabajará a partir de las variables del cuerpo del programa principal. No leerá nada desde el teclado.
3. Escribe los siguientes subprogramas
 - a) *precon_num_pista*, para comprobar que se cumple la precondition de que el número de pista esté entre 1 y 36, ambos inclusive.
 - b) *precon_distancia*, para comprobar que se cumple la precondition de que la distancia esté entre 0.5 y 5, ambos inclusive.
 - c) *precon_velocidad*, para comprobar que se cumple la precondition de que la velocidad está entre 90 y 150, ambos inclusive.
 - d) *calcula_headings*, que reciba un número de pista y devuelva los enteros *head1*, *head2*, *head3*, *head4*.
 - e) *calcula_tiempo*, que reciba distancia y velocidad y devuelva el tiempo. Referido todo al tramo de viento en cola.

Diseña estos subprogramas de la forma que estimes adecuada, el profesor no te dará indicaciones adicionales.

4. Escribe, además, todos los subprogramas necesarios para que el esqueleto haga lo que tiene que hacer, sin borrar nada.
5. En los subprogramas que comprueben precondiciones, declara los extremos de los rangos (1 y 36, 0.5 y 5, 90 y 150) como constantes locales de cada subprograma.
6. Recuerda que si el programa no compila, su nota será prácticamente nula.

Solución

```
{ $mode objfpc } { $H- } { $R+ } { $T+ } { $Q+ } { $V+ } { $D+ } { $X- } { $warnings on }

program circuito;

function precon_num_pista(num_pista:integer): boolean;
const
  Pista_min = 1;
  Pista_max = 36;
begin
  result := (num_pista >= Pista_min) and (num_pista <= Pista_max)
end;

function precon_distancia(distancia:real):boolean;
Const
  Distancia_minima = 0.5;
  Distancia_maxima = 5;
begin
  result := (distancia >= Distancia_minima) and (distancia <= Distancia_maxima);
end;

function precon_velocidad(velocidad:integer):boolean;
Const
  Velocidad_minima = 90;
  Velocidad_maxima = 150;
begin
  result := (velocidad >= Velocidad_minima) and (velocidad <= Velocidad_maxima)
end;

function calcula_tiempo(distancia:real; velocidad:integer):real;
// Recibe distancia en millas, velocidad en nudos.
// Devuelve tiempo en segundos.
begin
  result := 3600* (distancia / velocidad);
end;

function resta_90(heading:integer):integer;
begin
  heading := heading-90;
  if heading < 1 then
    heading := heading + 360;
  result := heading;
end;

procedure calcula_headings(num_pista: integer; var head1,head2,head3,head4:integer);
begin
  head1 := num_pista * 10;
  head2 := resta_90(head1);
  head3 := resta_90(head2);
  head4 := resta_90(head3);
end;

procedure escribe_entrada(num_pista, velocidad: integer; distancia: real);
begin
  write('Número de pista: ', num_pista);
  write(chr(9)); // Tabulador
  write('Velocidad: ', velocidad, ' nudos');
```

```

    write(chr(9)); // Tabulador
    writeln('Distancia: ', distancia:0:1, ' millas');
end;

procedure escribe_instrucciones(head1, head2, head3, head4: integer;
    tiempo_v_cola, tiempo_v_cruzado: real);
begin
    writeln('Tramo inicial: ', head1, 'º. ');

    write('Tramo viento cruzado: ', head2, 'º. ');
    writeln(tiempo_v_cruzado:0:0, ' segundos');

    write('Tramo viento en cola: ', head3, 'º. ');
    writeln(tiempo_v_cola:0:0, ' segundos');

    write('Tramo base: ', head4, 'º. ');
    writeln(tiempo_v_cruzado:0:0, ' segundos');

    writeln('Tramo viento de frente: ', head1, 'º. ');
end;

procedure calcula_y_escribe(num_pista, velocidad: integer;
    distancia:real);
var
    head1, head2, head3, head4:integer;
    tiempo_v_cola, tiempo_v_cruzado: real;
begin
    escribe_entrada(num_pista, velocidad, distancia);
    if not precon_num_pista(num_pista) then
        writeln('Número de pista fuera de rango');

    if not precon_velocidad(velocidad) then
        writeln('Velocidad fuera de rango');

    if not precon_distancia(distancia) then
        writeln('Distancia fuera de rango');

    if precon_num_pista(num_pista) and precon_velocidad(velocidad) and
        precon_distancia(distancia) then begin

        tiempo_v_cola := calcula_tiempo(distancia, velocidad);
        tiempo_v_cruzado := tiempo_v_cola / 2;

        calcula_headings(num_pista, head1, head2, head3, head4);
        escribe_instrucciones(head1, head2, head3, head4, tiempo_v_cola, tiempo_v_cruzado);
    end;

    writeln;
end;

var
    num_pista, velocidad : integer;
    distancia : real;
begin
    num_pista := 18;
    velocidad := 120;
    distancia := 4;
    calcula_y_escribe(num_pista, velocidad, distancia);

    num_pista := 10;
    velocidad := 100;
    distancia := 2;
    calcula_y_escribe(num_pista, velocidad, distancia);

    num_pista := 40;
    calcula_y_escribe(num_pista, velocidad, distancia);

    num_pista := 18;
    velocidad := 89;
    calcula_y_escribe(num_pista, velocidad, distancia);

    velocidad := 151;
    calcula_y_escribe(num_pista, velocidad, distancia);

```

```
velocidad := 110;  
distancia := 0.4;  
calcula_y_escribe(num_pista, velocidad, distancia);  
  
velocidad := 110;  
distancia := 5.5;  
calcula_y_escribe(num_pista, velocidad, distancia);  
end.
```