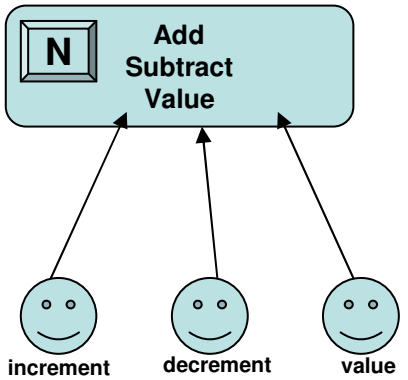
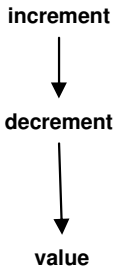
Como é, neste caso, feita a protecção ao dado partilhado?

```
with Ada.Integer_Text_Io;
use Ada.Integer_Text_Io;

procedure Dado_Partilhado is

   A:Integer;

   task Access_Control is
      entry Add (value: in Integer);
      entry Subtract (Value: in Integer);
      entry Value (Value:out Integer);
   end Access_Control;

   task body Access_Control is
      N:Integer:=0;
   begin
      loop
         accept Add (Value:in Integer) do
            N:=N+value;
         end Add;
         accept Subtract (Value: in Integer) do
            N:=N-value;
         end Subtract;
         accept Value (Value:out Integer) do
            Value:=N;
            end;
         end loop;
      end access_control;
```

```
task Decrement;
      task body Decrement is
      begin
         Access_Control. Subtract(1);
      end Decrement;

      task Increment;
      task body Increment is
      begin
         Access_Control. Add (1);
      end increment;

      begin
         Access_Control.Value(a);
         Put(a);
      end Dado_Partilhado;
```





Problemas???

Um atraso da tarefa Increment...

Altere o código por forma a permitir que qualquer das tarefas increment e decrement possa ser a primeira a alterar a varável protegida...

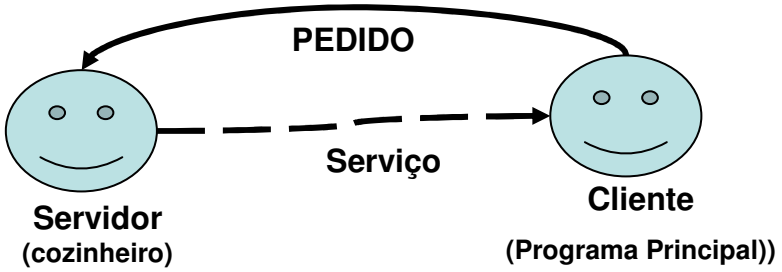
mas garanta que o resultado final é consistente...

Crie um guarda na tarefa decrement que garanta que n nunca terá o valor -1.

Introduza atrasos de 2 segundos nas tarefas increment e decrememement e altere o tratamento select anteriormente criado por forma a que seja possível enviar, em intervalos de 0,5 segundos, a mensagem "nenhum pedido" para o monitor.

Crie uma forma do programa terminar a sua execução...

```
with Ada.Text_Io, Ada.Integer_Text_Io;
use Ada.Text_Io, Ada.Integer_Text_Io;

procedure Cachorro_Quente is

   task Cozinheiro is
      entry Faz_Cachorro(Numero : Integer);
   end Cozinheiro;

   procedure Pedido(Numero:Integer) renames
      Cozinheiro.Faz_Cachorro;

   task body Cozinheiro is
   begin
      Put_Line("Estou pronto para servir um cachorro");

      --mas só o serve se alguem o pedir.... não faz cachorros
antes de serem pedidos

      accept Faz_Cachorro(Numero : Integer) do
         Put_Line("vai sair o primeiro cachorro");
         Put("estou a confeccionar o cachorro ");
         Put_Line("um pouco de mostarda");
         delay 2.0;
      end Faz_Cachorro;

      --espera pedidos de mais 4 cachorros
for Index in 1..4 loop

         accept Faz_Cachorro(Numero : Integer) do
            Put("Vai sair o cachorro numero");
            Put(Numero, 3);
            New_Line;
            Put("estou a confeccionar o cachorro ");
            Put_Line("um pouco de mostarda");
            delay 2.0;
         end Faz_Cachorro;
      end loop;
```




```
   accept Faz_Cachorro(Numero : Integer) do
         Put("Vai sair o ultimo cachorro");
         New_Line;
         Put("estou a confeccionar o cachorro ");
         Put_Line("um pouco de mostarda");
         delay 2.0;
      end Faz_Cachorro;

      Put_Line("Acabaram-se os cachorros");
   end Cozinheiro;

begin
   for Index in 1..5 loop
      Put_Line ("cliente pede o cachorro");
      Pedido(Index);
      Put_Line("o cliente esta a comer o cachorro");
      New_Line;
   end loop;
      Put_Line ("cliente pede o cachorro");
      Cozinheiro.faz_cachorro(6);
      Put_Line("o cliente esta a comer o cachorro");
      New_Line;
   Put_Line("Não tenho mais fome");
end Cachorro_Quente;
```

Aumente o número de pedidos do cliente...

Resolva a situação criada recorrendo a uma excepção (tasking_error).

Substitua o cliente "programa principal" por dois clientes tarefa, o Joao e o Pedro...

O Joao terá de comer 2 cachorros e o Pedro 3

Substitua as tarefas Pedro e Joao por tipos tarefa.

Lance dinâmicamente as tarefas Pedro e Joao por forma a que o programa faça o mesmo que anteriormente.