Concurso/Encontro Nacional de Programação Lógica CNPL'98

Dep. de Matemática, Univ. de Aveiro

27–28 de Abril de 1998

CODIFICA

Codificador Primo

Consideremos os números naturais e a ideia muito simples de número primo: um número que não se pode obter pela multiplicação de dois números mais pequenos onde não se inclui a unidade. Sabia-se desde tempos remotos, como um facto empírico, que todos os números se exprimem, de forma única, como um produto de primos; e os Gregos conseguiram prová-lo. Algumas perguntas sobre primos surgem naturalmente. Por exemplo: Quantos primos existem e como estão os primos distribuídos entre os outros naturais? Como se determinam primos? Como é que se mostra que um dado natural é, ou não, um primo? Como se factoriza, num produto de primos, um natural arbitrário? Mesmo com tão simples matéria-prima, a matemática fez uma obra espantosa e, se questões como as acima colocadas, podem ser olhadas numa perspectiva abstracta e como um ramo da matemática inaplicável, a verdade é que estas questões têm um significado crucial para a criptografia; tanto mais que têm havido sérias tentativas de classificar alguns dos resultados desta área como segredos militares. Tudo indica que são os métodos de codificação baseados em números primos que constituirão a parte central do sistema de segurança da futura auto-estrada da informação.

Tarefa

Implemente em Prolog um codificador de mensagens e o respectivo descodificador. O codificador deve substituir o código ASCII de cada caracter da mensagem por um outro de acordo com o algoritmo de encriptação descrito pelos seguintes passos:

- Converter a mensagem dada numa lista L com os códigos ASCII de cada caracter.[1]

[1] Sugere-se o uso do predicado pré-definido `name/2`.

- Gerar a lista P de todos os primos até ao código ASCII do primeiro caracter da frase a codificar (1^o elemento de L), usando um predicado `primos/2` que terá de desenvolver segundo o método a indicar mais adiante.
- O primeiro caracter da frase não é alterado.
- Ao segundo caracter somamos o primeiro número primo em P; ao terceiro caracter somamos o segundo elemento de P, etc, considerando a lista P de um modo circular —depois de somar o último primo em P, voltar ao início da lista.

Lembramos que o código ASCII é um número entre 0 e 255 e que é preciso ter o cuidado de garantir que os códigos ASCII codificados também estão nesta gama! Mais uma vez, deve usar a noção de circularidade: por exemplo, se estamos a somar o primo 3, o código ASCII 33 será transformado no 36 enquanto o 254 será transformado no 1.

O algoritmo de descodificação deverá proceder à operação inversa do de codificação.

Subtarefa

Implemente um predicado em Prolog para gerar números primos segundo o algoritmo designado por *Crivo de Eratóstenes*. Esse método permite-nos construir uma lista de todos os primos até um limite dado, de acordo com a seguinte descrição[2]:

> *"Escrevemos os naturais desde* 2 *até ao limite desejado; por exemplo* 200*:*
>
> 2, 3, 4, 5, 6, 7, ..., 198, 199, 200.
>
> *Começando no princípio da lista, o primeiro número que encontramos é* 2*, um primo. Deixamos* 2 *de lado, e passamos à frente marcando os números de* 2 *em* 2*, isto é, marcamos* 4, 6, 8, 10, $\cdots$*. Depois de ter marcado todos os números pares até* 200*, voltamos ao princípio da lista para encontrar o primeiro número depois de* 2 *que não foi marcado: é* 3*. Deixamos* 3 *de lado (é primo), e passamos à frente marcando os números de* 3 *em* 3*: marcamos* 6, 9, 12, 15, $\cdots$*. Prosseguimos assim. Depois de fazermos este jogo repetidamente para* 2, 3, ..., N*, onde* N *é o maior natural não marcado tal que* $N \leq \sqrt{200}$*, então os números da lista que ainda não foram marcados são os primos até* 200[3]*."*

Os Dados

O único dado necessário para a resolução da tarefa principal deste problema é a mensagem a codificar. Essa mensagem será um parâmetro do predicado `encode/2` que tem de desenvolver, como se explica na secção seguinte.

[2] B. Owen, *Teoria Elementar dos Números I*, Boletim da SPM, 33, (1995), p. 17-36.

[3] Sugerimos que evitem o cálculo da raiz quadrada. Na situação apresentada, o teste $N \leq \sqrt{200}$ pode ser substituído por $N^2 \leq 200$.

Para a subtarefa de geração de números primos também precisa de conhecer à partida apenas um dado: o valor do limite superior da lista a gerar.

Os Resultados

O seu programa deve ser invocado através dos predicados:

a) `encode/2`

que recebe a mensagem e retorna no segundo argumento a lista dos códigos ASCII da mensagem codificada

b) `decode/2`

que recebe a lista dos códigos ASCII da mensagem codificada e devolve no segundo argumento a mensagem original.

c) `primos/2`

que recebe como primeiro argumento um número natural N e devolve no segundo argumento a lista de todos os primos até N.

Exemplos:

```
?- primos(100,L).
  L = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,
       67,71,73,79,83,89,97]

?- encode('Ola malta',C), decode(C,D).

  C = [79,110,100,37,116,108,121,133,116]
  D = 'Ola malta'
```

Avaliação

Cada um dos predicados `primos/2`, `encode/2` e `decode/2`, será testado com 2 exemplos. Para cada teste executado com sucesso recebe 15 pontos, exceptuando os dois testes de `decode/2` que valerão 10 pontos cada um. Os restantes 20 pontos serão atribuídos em função da análise do seu código.
Em caso de necessidade de desempate com outras equipas,[4] será tido em consideração o maior número para o qual o seu predicado `primos/2` é capaz de gerar, sem problemas, a lista de primos.

[4] relevante apenas para os primeiros três lugares – cf. "Regulamento do CNPL".