

# Pesquisa de instruções

A equipe do Fórum de GCC e Assemblers (FGA) está desenvolvendo uma ferramenta de engenharia reversa que, dado um código em linguagem de máquina, deseja convertê-lo novamente num código em linguagem de alto nível (alguma das quais o GCC dá suporte). Há inúmeros módulos que precisam ser desenvolvidos, e você faz parte da equipe de desenvolvimento. O gerente do projeto te designou a seguinte tarefa: você tem que desenvolver um programa que lê o código de uma instrução em linguagem de máquina e devolve a correspondente da linguagem de alto nível. Por exemplo, se o seu programa ler o código 20, ele deve devolver o **for**. Como você acabou de aprender um algoritmo de busca bem eficiente, você decide fazer um programa que carrega na memória todas as instruções, e seus respectivos códigos, e depois responde às pesquisas demandadas ao seu programa.

## Entrada

A entrada é composta de diversas linhas. A primeira linha contém um inteiro  $n$  que determina quantas instruções você precisa carregar na memória. As  $n$  linhas seguintes contém um inteiro e uma palavra que caracterizam uma instrução: o primeiro é o código da instrução, e a segunda a palavra-chave com, no máximo, 15 caracteres. As demais linhas são inteiros que correspondem aos códigos de instruções para os quais você precisa determinar a palavra-chave correspondente. A entrada termina com EOF.

## Saída

A saída é composta pela mesma quantidade de consultas feitas na entrada. Para cada inteiro consultado, você deve imprimir a palavra-chave correspondente. Se a palavra chave não fizer parte do conjunto de instruções, você deve imprimir **undefined**.

## Exemplo de entrada

```
5
220 while
954 switch
344 for
104 if
858 case
344
468
104
220
948
732
858
564
104
954
```

## Exemplo de saída

```
for
undefined
if
while
undefined
undefined
case
undefined
if
switch
```

*Author: John L. Gardenghi*