Problem 1: Dot Product

The dot product of two vectors, \( u \) and \( v \), is \( \sum_i u_i * v_i \).

Examples

Write two examples of the operation of dot-product.

Implementation

Write a function \texttt{dot-product} that computes the dot-product of two vectors.

\begin{verbatim}
(define (dot-product u v)
  ;; dot-product: (vectorof number) (vectorof number) -> number

  ...)
\end{verbatim}

Test

Demonstrate the operation of your function on the examples you defined above.

Problem 2: Changeable Phonebook

Assume a variant of the phonebook in Homework 6, where instead of a list of structures, the phonebook is represented as a vector of structures as below.

A phone-book is a vector of length 100 where entries are either:

- \#f, or
- \texttt{(make-pb name number)}

\begin{verbatim}
(define-struct pb (name number)), where name is a symbol and number is a number
\end{verbatim}

\texttt{new-phonebook}

Based on the definition above, create a new phonebook where all the entries are \#f.

\texttt{add-phonebook}

Create a new function \texttt{add-phonebook} that inserts new phonebook entry - name and number - into the phonebook created above. If an entry already exists for a given name, do nothing.

\begin{verbatim}
(define (add-phonebook name number)
  ;; add-phonebook: symbol number -> (void)

  ...)
\end{verbatim}
update-phonebook

Implement a new function **update-phonebook** that takes a name and number and updates the associated phonebook entry if there is one, and returns #f, otherwise.

```scheme
(define (update-phonebook name number)
  ;; add-phonebook: symbol number -> (void) or \#f
```