目录

Week 1 Exercises

fiveDigit.c

There is a 5-digit number that satisfies 4 * abcde = edcba, that is,when multiplied by 4 yields the same number read backwards.Write a C-program to find this number.

innerProdFun.c

Write a C-function that returns the *inner product* of two *n*-dimensional vectors**a** and **b**, encoded as 1-dimensional arrays of *n* floating point numbers.

Use the function prototype **float innerProduct(float a[], float b[], int n)**.

By the way, the inner product of two vectors is calculated by the sum for $i=1..n$ of $a_i * b_i$

matrixProdFun.c

Write a C-function to compute **C** as the *matrix product* of matrices **A** and **B**.

Use the function prototype **void matrixProduct(float a[M][N], float b[N][P], float c[M][P])**

You can assume that *M, N, P* are given as symbolic constants, e.g.

```
#define M 3
#define N 4
#define P 4
```

By the way, the product of an *m x n* matrix **A** and an *n x p* matrix **B** is the *m x p* matrix **C** such that $\mathbf{C_{ij}}$ is the sum for $k=1..n$ of $\mathbf{A_{ik}} * \mathbf{B_{kj}}$ for all $i \in \{1..m\}$ and $j \in \{1..p\}$

able.c

Write a C-program that outputs, in alphabetical order, all strings that use each of the characters 'a', 'b', 'l', 'e' exactly once.

How many strings are there actually?

collatzeFib.c

a. Write a C-function that takes a positive integer *n* as argument and prints a series of numbers generated by the following algorithm, until 1 is reached:
   - if *n* is even, then $n \leftarrow n/2$
   - if *n* is odd, then $n \leftarrow 3*n+1$

   (Before you start programming, calculate yourself the series corresponding to *n=3*.)

b. The Fibonacci numbers are defined as follows:
   - Fib(1) = 1
   - Fib(2) = 1
   - Fib(*n*) = Fib(*n*-1)+Fib(*n*-2) for $n \geq 3$

   Write a C program that calls the function in Part a. with the first 10 Fibonacci numbers. The program should print the Fibonacci number followed by its corresponding series. The first 4 lines of the output is as follows:

```
Fib[1] = 1:
Fib[2] = 1:
Fib[3] = 2: 1
Fib[4] = 3: 10 5 16 8 4 2 1
```

fastMax.c

Write a C-function that takes 3 integers as arguments and returns the largest of them. The following restrictions apply:

- You are permitted to only use assignment statements, a return statement and Boolean expressions
- You are not permitted to use if-statements, loops (e.g. a while-statement), function calls or any data or control structures

(You might find this exercise a 'brain-twister'.)

Week1Exercises (2019-06-08 09:51:29由AlbertNymeyer编辑)