

Sparse Vector Arithmetic

In this lab, you will implement an interpreter for sparse vector operations. The interpreter uses link lists for representing sparse vectors. A vector is a one-dimensional array of elements, so a natural way to represent a vector in C is through an array. But for vectors that have many elements of which most are zero (i.e. sparse vectors), the array implementation is a waste of space; rather a linked list representation is used.

1. Representation of a Sparse Vector

A sparse vector is represented as a structure containing its size and a pointer to its elements. The elements of a vector are stored in a linked list structure. The link list contains only those elements that are not zero and stores an element together with its index (its position in the vector).

```
struct el{
    int index;
    int data;
    struct el * elements;
};

typedef struct el eltype;

struct svector{
    int size;
    eltype * elements;
};
```

2. The Interpreter

Your program will work like an interpreter: it will type a prompt `>` on the screen and wait for user input. The user input will consist of single lines. Your program will read this line and perform the operations it requests followed by an output printed on the screen.

2.1 Vector Database

Your program will read a database of sparse vectors from a text file. Each vector is represented as decimal values placed on a separate line. For example, the following input file:

```
4.3 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 5 45.2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

represents the two vectors:

```
Vector1 : (4.3, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 45.2)
Vector2 : (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
```

The file may contain at most 100 vectors. The vectors will be referred to with the keyword `v` and their order in the database file.

2.2 Operations on Sparse Vectors

The operations on vectors that you will be implementing are:

- addition (+)
- subtraction (-)
- product with scalar and scalar product (.). You can choose another representation of the operator.

Additionally, it will be possible to assign results of operations to existing vectors or the special vector **temp** and print vectors. Each user command can contain at most one operation and at most one assignment. It is enough to type the name of a vector on a single line to print its value. Finally, one can quit by typing **q**.

2.3 Example Run

```
Welcome to SparseLab!
Database read, waiting for command
> v1
v1 is 4.3 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 5 45.2
> v2
v2 is 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> v2 = v1 + v2
v2 = 4.3 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 5 45.2
> temp = 2 . v2
temp is 8.6 0 0 0 0 0 0 16 0 0 0 0 0 0 0 0 0 10 90.4
> v1 . v1
2150.53
>q
Bye bye!
```

3 Demos

Demonstrations will be done during lab hours. You will find test cases at the home page for you to try before you demo.