w3e26.h

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Header file

#ifndef W3E26FUNCTIONS
#define W3E26FUNCTIONS

#include <iostream>
#include <string>

enum actions
{
  buy = 1, sell, smthelse // Starting at 1 to avoid problems with using 0
};

enum Constants
{
  BitsPerValue = 2,  // Var 'b'
  ValuesPerByte = (sizeof(uint8_t) * 8) / BitsPerValue
  // Coincides with one byte, but is catered to variable instead
};

size_t withinIndex (size_t elementN);
size_t arrayIndex (size_t elementN);
size_t getField (const uint8_t *byteArray, size_t elementN);

#endif
```

main.ih

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Main file: internal header

#include <iostream>
#include <string>
#include <cstdint>
#include <fstream>

using namespace std;

size_t arraySize (size_t numOperations);
size_t numOps (const std::string fileAsString);
void setBits (uint8_t *byteArray, size_t numOps, const std::string fileAsString);
void show (const uint8_t *byteArray, size_t numOperations);
```

main.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Main file

#include "main.ih"

int main()
{
  cout << "Specify the file that you want to input. \n";

  string fileName;      // Initialise string for name input file
  cin >> fileName;      // Populate filename string
  ifstream inputFile(fileName.c_str()); // Convert filename to cstring and input it

  string fileAsString;  // Initialise string containing input file as string
  getline(inputFile, fileAsString); // Move the input file to the string

  size_t numOperations = numOps(fileAsString);  // Number of actions in file ('n')

  uint8_t byteArray[arraySize(numOperations)]; // Initialise array of req. size
  setBits(byteArray, numOperations, fileAsString); // Populate it

  show(byteArray, numOperations); // Print it
}

// Example file randVals.txt was used, containining a random assortment of:
// B, S, E 33% each
// Rest N/A or -

```

arrayIndex.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Function: get array index corresponding to element N

#include "w3e26.h"

size_t arrayIndex (size_t elementN)
{
  return elementN / ValuesPerByte;  // Integer division
};
```

arraySize.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Function: determines necessary array size

#include "w3e26.h"

size_t arraySize (size_t numOperations)
{
  return ((numOperations) + (numOperations - 1)) / ValuesPerByte;
  // Returns requisite array size given n and k
};
```

getField.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Function: retrieves element from array

#include "w3e26.h"

size_t getField (const uint8_t *byteArray, size_t elementN)
{
  size_t action = (byteArray[arrayIndex(elementN)]
                     >> withinIndex(elementN) * BitsPerValue
                     &
                     ( (1 << BitsPerValue) - 1 ) );
  return action;
  // Returns action stored at given location in given array
};
```

numOps.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Function: determine number of values in file

#include "w3e26.h"

size_t numOps (const std::string fileAsString)
{
  size_t numOperations = 0;
  for (size_t index = 0; index != fileAsString.length(); ++index)
  // Loop through all characters of the string
  {
    if (fileAsString[index] == ',') // If it comes across a comma
      ++numOperations;  // Increment the counter
  }
  return numOperations + 1; // Return the counter, plus one for the final element
};
```

setBits.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Function: populate character array with bits from input file

#include "w3e26.h"

void setBits (uint8_t *byteArray, size_t numOps, const std::string fileAsString)
{
  size_t elementCounter = 0, last = 0, next = 0;  // Init counters
  char delimiter = ','; // Delimiter to find seperate actions

  while (elementCounter != numOps + 1) // While counter is not through list yet
  {
    next = fileAsString.find(delimiter, last);  // Find next delimiter position
    std::string subString = fileAsString.substr(last, next - last);
    // Create substring based on previous and newly found delimiter position

    last = next + 1;
    // Set start next action to just after previously found delimiter

    size_t byteIdx = arrayIndex(elementCounter);
    size_t withinIdx = withinIndex(elementCounter);
    // Find position in array for action, both element and within-element index

    if (subString.find("B") != std::string::npos)
      byteArray[byteIdx] |= buy << withinIdx * BitsPerValue;
    else if (subString.find("S") != std::string::npos)
      byteArray[byteIdx] |= sell << withinIdx * BitsPerValue;
    else
      byteArray[byteIdx] |= smthelse << withinIdx * BitsPerValue;
    // Set appropriate position within array to match substring

    ++elementCounter; // Increment element counter
  }
};
```

show.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Function: print entire bit array

#include "w3e26.h"

void show (const uint8_t *byteArray, size_t numOperations)
{                    // For all stored actions
  for (size_t index = 0; index != numOperations; ++index)
  {
    switch (getField(byteArray, index)) // Output associated character
    {
      case buy:
        std::cout << 'B';
        break;
      case sell:
        std::cout << 'S';
        break;
      case smthelse:
        std::cout << 'E';
        break;
      default:
        std::cout << "\nSomething is broken. \n";
        break;
    }
    std::cout << ",";
  }
};
```

withinIndex.cc

```
// Programming in C/C++
// Week 3: Assignment 26
// Tjalling Otter & Emiel Krol
// Function: get within array element index of element N

#include "w3e26.h"

size_t withinIndex (size_t elementN)
{
  return elementN % ValuesPerByte;
  // Returns the within within element index of a given value
};
```