# C5 – Data Structures and Dynamic Memory Allocation

Lab Work

3.1 Vectors

```
Vector createVector(const unsigned int nLength)
{
    Vector vec;
    vec.length = nLength;
    vec.element = (double *) malloc(nLength * sizeof(double));
    return vec;
}
```

```
void destroyVector(Vector vec)
{
    free(vec.element);
}
```

```
#include <stdio.h>
#include "vector.h"


#define VEC_FILE "example1.vec"


Vector v;


int main() {
    v = createVectorFromFile(VEC_FILE);

    printVector(v);

    destroyVector(v);

    return 0;
}
```

```
0] = 2.300000
1] = 4.500000
```

example1.vec

```
2.3 4.5
```

## 3.2 Matrices

```
Matrix createMatrix(const unsigned int nRows, const unsigned int nCols)
{
    Matrix mat;
    mat.rows = nRows;
    mat.cols = nCols;

    mat.element = (double **) malloc(nRows * sizeof(double *));

    for(int i=0; i<nRows; i++) {

        mat.element[i] = (double *)malloc(nCols * sizeof(double));
    }
    return mat;
}
```

```
void destroyMatrix(Matrix mat)
{
    for(int i=0; i<mat.rows; i++) {
        free(mat.element[i]);
    }
    free(mat.element);
}
```

```
#include <stdio.h>
#include "vector.h"
#include "matrix.h"


#define VEC_FILE "example1.vec"
#define MAT_FILE "example1.mat"


Vector v;
Matrix m;

int main() {
    v = createVectorFromFile(VEC_FILE);
    m = createMatrixFromFile(MAT_FILE);


    //printVector(v);
    printMatrix(m);

    destroyVector(v);
    destroyMatrix(m);

    return 0;
}
```

```
[0][0] = 2.300000
[0][1] = 4.500000
[1][0] = 2.400000
[1][1] = 6.300000
```

example1.mat

```
2.3 4.5
2.4 6.3
```

## 3.3 Circuit Simulation

```c
typedef enum {
    resistor,
    voltage,
    current,
} CompType;

typedef struct {
    unsigned int n1;
    unsigned int n2;
    double value;
    CompType type;
    char name[32];
} Component;

typedef struct {
    unsigned int nV;
    unsigned int nI;
    unsigned int nR;
    unsigned int nC;
    unsigned int nN;
    Component *comp;
} Circuit;
```

```c
#include <stdio.h>
#include <stdlib.h>
#include "circuit.h"

int main(int argc, char *argv[]) {
    Circuit c;

    if (argc == 2) {
        c = createCircuitFromFile(argv[1]);

        analyseCircuit(c);

        destroyCircuit(c);
    }
    else
        printf("Syntax: %s <filename>\n", argv[0]);

    return EXIT_SUCCESS;
}
```

```
Circuit Analysis----------------------------
 Voltage sources: 1
 Current sources: 1
       Resistors: 6
           Nodes: 5
-----------------------------
 Node   0 =   0.000000 V
 Node   1 =   1.329398 V
 Node   2 =   1.000000 V
 Node   3 =   0.000000 V
 Node   4 =   1.328829 V
-----------------------------
 I(V1)    =   0.000356 A
-----------------------------
```