ECEN 449 Homework 2

Philip Smith - 624002014 (Sec. 511)

February 26, 2020

Problem 1

Below is the code for problem 1:

```c
#include <stdio.h>
#include <string.h>

//populate_matrix
//This function takes the matrix object and inserts data from
//the corresponding file pointer.
void populate_matrix(int r, int c, FILE *data, float mat[r][c]){
    float readval; //temp storage for read value

    //Loop across all elements of the matrix
    for (int i = 0; i < r; ++i){
        for (int j = 0; j < c; ++j){
            fscanf(data,"%f",&readval); //Scan for the data
            mat[i][j] = readval; //Set the data into the matrix
        }
    }
}

//write_matrix
//This function writes a matrix to a file
void write_matrix(int r, int c, float mat[r][c], FILE *o){
    fprintf(o,"%d %d\n",r,c); //Print rows and columns
    //Then print the matrix elements
    for (int i = 0; i < r; ++i){
        for (int j = 0; j < c; ++j){
            fprintf(o,"%f ",mat[i][j]);
        }
    }
}

//multiply_matrices
//This function takes 2 input matrices and multiplies them together
//into a third matrix.
void multiply_matrices(int ah, int aw, int bh, int bw, int ch, int cw, float a[ah][aw],
    float b[bh][bw], float c[ch][cw]){

    for (int i = 0; i < ah; i++){
        for (int j = 0; j < bw; j++){
        c[i][j] = 0; //Pre-set the value of the output matrix
            for (int k = 0; k < bh; k++){
                c[i][j] += a[i][k]*b[k][j]; //Compute summation
            }
        }
    }
}

int main() {
    FILE *m1,*m2,*out; //File pointers
    int aw,ah,bw,bh,cw,ch; //Matrix A/B/C (w)idth/(h)eight

    //Open all associated files
    m1 = fopen("./inA.txt", "r");
    m2 = fopen("./inB.txt", "r");
    out = fopen("./outC.txt","w");

    //Get the dimensions of the supplied matrices
    //and place this info into int variables
    fscanf(m1,"%d %d",&ah,&aw);
    fscanf(m2,"%d %d",&bh,&bw);

    //Check if we can perform the multiplication
    if(bh != aw){
        //Cannot perform multiplication. Quit!
        printf("Error: Matrix dimensions incompatible!");
        return 1;
    }else{
        //Set the dimensions for the output matrix
        ch = ah;
        cw = bh;
    }

    //Pre-allocate memory for all matrices
    //NOTE: They will not have valid data in them!
    float matA[ah][aw];
    float matB[bh][bw];
    float matC[ch][cw];

    //Populate the matrices with data from files
    populate_matrix(ah,aw,m1,matA);
    populate_matrix(bh,bw,m2,matB);

    //Multiply matrices A and B into C
    multiply_matrices(ah,aw,bh,bw,ch,cw,matA,matB,matC);

    //Write the subsequent value of C to a file
    write_matrix(ch,cw,matC,out);

    //Close files before exit
    fclose(m1);
    fclose(m2);
    fclose(out);

    return 0;
}
```

The following is the result from calculation A, which I confirmed was correct (ignoring the floating-point errors). There was nothing printed to stdout so I have pasted the resultant file:

```
$ cat outC.txt
2 2
11.800000 27.779999 24.549999 56.039997
```

The following is the result from calculation B, which returned an error:

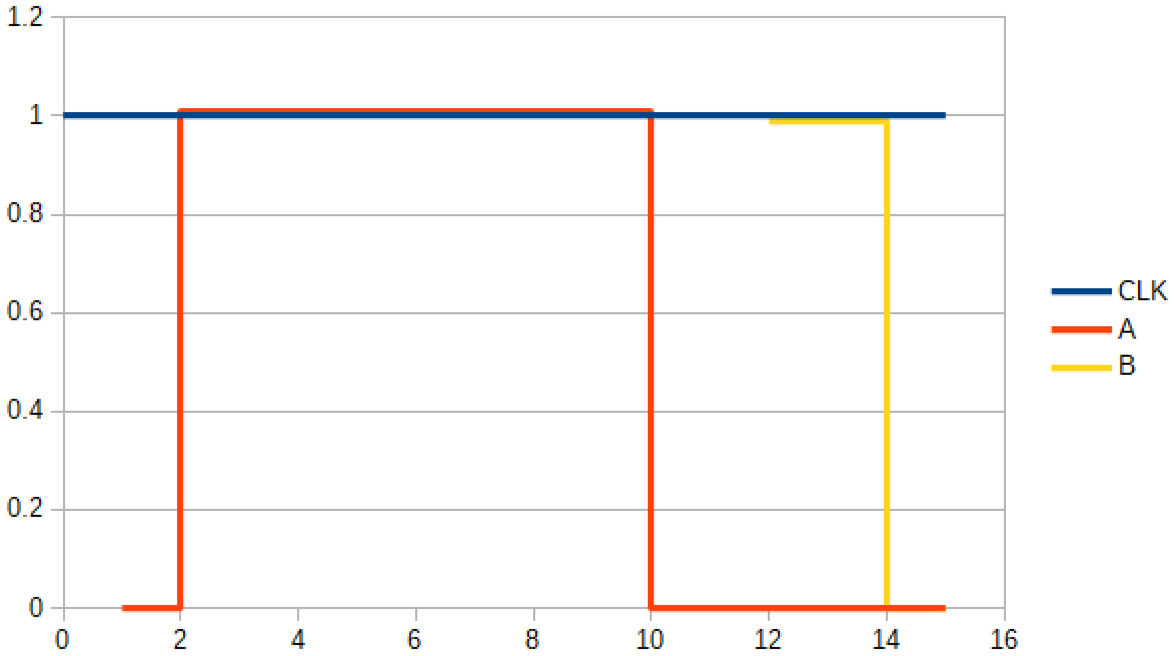
```
$ ./a.out
Error: Matrix dimensions incompatible!
```

Problem 2

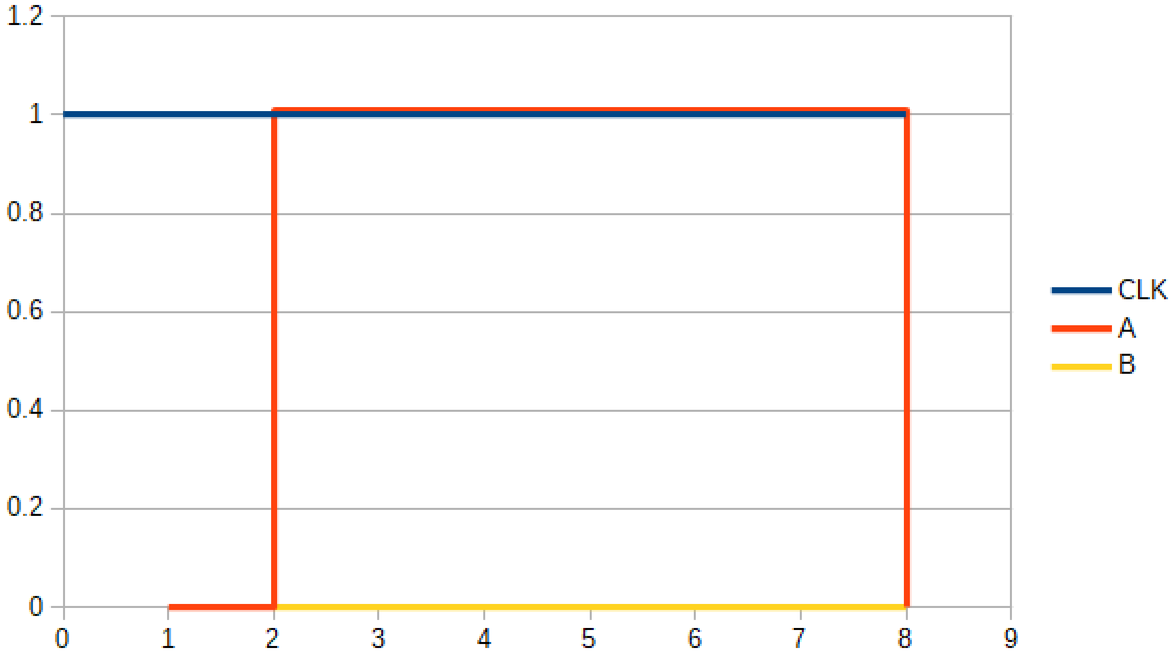
a) The following files are created: block1.o, block2.o, final, and main.o.
b) If we modify block1.c and issue another make, final and block1.o will be regenerated. This is because the program that compiles to block1.o was changed, requiring a recompilation. In addition, the output binary ”final” also depends on block1.o, meaning it too must be recompiled.
c) No new files will be created.

Problem 3

The following is my plot for code fragment A:



The following is my plot for code fragment B:



The above charts will not display any data when the values for a is X or b is X, respectively.