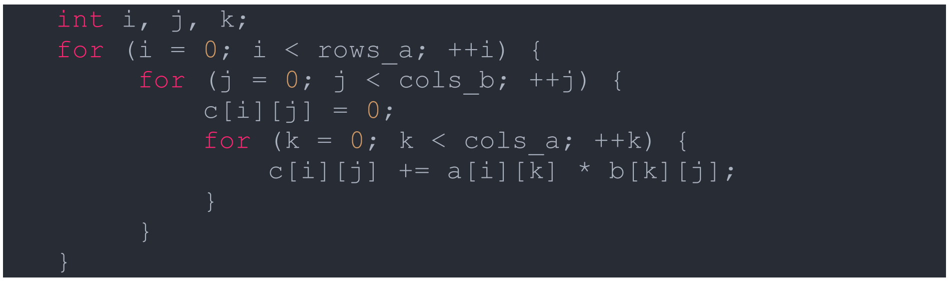
Akdeniz University Computer Engineering Department
CSE206 – Computer Organization (2022-2023 Spring)
Homework #01

HW01 – The Effect of Row Major Access in Matrix Multiplication Problem

Due Date: Tuesday, 25 April 2023, @23:59 pm (midnight)

Considering the following 2D matrix multiplication problem, improve the traditional algorithm by considering the row major access principles.



```
int i, j, k;
for (i = 0; i < rows_a; ++i) {
     for (j = 0; j < cols_b; ++j) {
          c[i][j] = 0;
          for (k = 0; k < cols_a; ++k) {
               c[i][j] += a[i][k] * b[k][j];
          }
     }
}
```

The following code is an example code we wrote in the lecture.

```
//Sample code for matrix addition
//Header libraries
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
//Main function
int main(int argc,char *argv[]){
 //argc used for the number of arguments passed
 //argv[] used for the values of the arguments
 struct timeval start,end; //for time measurements
 int row,col,i,j; //array dimensions
 int **a,**b,**c; // double pointers
 row=atoi(argv[1]); //ascii to integer conversion
 col=row;
 //double pointer allocations for each row of the array
 a=(int **)malloc(sizeof(int *)*row);
 b=(int **)malloc(sizeof(int *)*row);
 c=(int **)malloc(sizeof(int *)*row);
 for(i=0;i<row;i++){
  //pointer allocations for each column of the array
  a[i]=(int *)malloc(sizeof(int)*col);
  b[i]=(int *)malloc(sizeof(int)*col);
  c[i]=(int *)malloc(sizeof(int)*col);
  for(j=0;j<col;j++){
    //Initializations
    a[i][j]=0;
    b[i][j]=rand(); //random numbers
    c[i][j]=rand();
  }
 }
 printf("Row major\n");
 gettimeofday(&start,NULL);
 for(i=0;i<row;i++){
  for(j=0;j<col;j++){
    a[i][j]=b[i][j]+c[i][j];
  }
 }
 gettimeofday(&end,NULL);
 printf("Time to compute: %4.4f seconds\n",(((end.tv_sec-
start.tv_sec)*1000000)+(end.tv_usec-start.tv_usec))/1000000.0);
 //Memory deallocation
 for(i=0;i<row;i++){
   free(a[i]);
   free(b[i]);
   free(c[i]);
  }
  free(a);
  free(b);
  free(c);
  return(0);
 }
```

Use the command to compile the sample code.

```
gcc -Wall -o sample.exe sample.c
```

To run the example code

```
#./sample.exe  5000
```

You should provide a complete C code that performs **2D matrix multiplication**. Then **compare** the timing results of **original algorithm** and your **modified version** that supports **row major ordering** for the row size values of 128, 256, 512, 1024, 2048, 4096, 8192, 16384, (If possible 32768) and measure the time (**wall clock time**) it takes to perform the **multiplication**.

Grading:

Dynamic memory allocation : 10 points (matrix size will be determined by a parameter)
Program parameters : 5 points (used to determine the matrix dimensions, row and column)
Code comments : 5 points (Commenting necessary code segments/lines)
Correct use of row major ordering: 45 points
Assignment report : 25 points (Cover page, Table of contents, Problem description, Methodology, Visual Results e.g. table figure etc.)
Comparison Figure : 10 points (Comparison of timing results for the original and modified multiplication algorithm)

Compile error : -20 points
Semantic error : -20 points
Compile warnings : -10 points
Filename format : -25 points (cse206-2023-[yourstudentid]-hw01.c)

Submission:

Upload your document cse206-2023-[yourstudentid]-hw01.c file to the Teams. For example, if it is 20230808123 then your filename must be cse206-2023-20230808123-hw01.c, otherwise, you will lose 25 points.

Late assignment submission policy : -20 points (Every day)