

Compilation: g++ -std=c++11 convol.cpp -fopenmp -o convol

Execution: ./convol 1024 768 3

2)

Flops needed to be done to compute a convolution of dimension k on an image of size $n \times m$
 = (no. of additions and multiplications \times matrix size \times total size of 2-D array)
 = $2 \times k \times k \times n \times m$

Memory needed to be done to compute a convolution of dimension k on an image of size $n \times m$
 = Read $k \times k$ matrix + Read array + Write array
 = (size of type) $\times k \times k$ + (size of type) $\times n \times m$ + (size of type) $\times n \times m$
 = $4 \times k \times k + 4 \times n \times m + 4 \times n \times m$ (Assuming 32 bit machine, float and int are 4 bytes)

From Assignment 1 and 2,

Maximum number of floating operations in Mamba = 1638 GFlops / sec

Maximum number of Integer operations in Mamba = 1228 GIops / sec

In Mamba, maximum Bandwidth or Bytes fetched at a rate of (68+68 GB/s) = 136 GigaBytes/sec

Time taken for convolution of dimension 3 on an image of 1024x768 = max(FlopTime, MemTime)

For floating point arithmetic, Time = Total flops / (max flops / sec)
 = $2 \times 3 \times 3 \times 1024 \times 768 / (1.7 \times 10^{12})$
 = 8.642 μ sec

For memory operations, Time = Total bytes / (max bytes / sec)
 = $4 \times (2 \times 1024 \times 768 + 3 \times 3) / (136 \times 10^9)$
 = 46.2 μ sec

Hence, maximum time is taken by memory operation, time taken would be 46.2 μ sec

Time taken for convolution of dimension 11 on an image of 1024x768 = max(FlopTime, MemTime)

For floating point arithmetic, Time = Total flops / (max flops / sec)
 = $2 \times 11 \times 11 \times 1024 \times 768 / (1.7 \times 10^{12})$
 = 116.18 μ sec

For memory operations, Time = Total flops / (max flops / sec)
 = $4 \times (2 \times 1024 \times 768 + 11 \times 11) / (136 \times 10^9)$
 = 46.2 μ sec

Hence, maximum time is taken by floating point operations, time taken would be 116.18 μ sec

Performance is measured in pixels/sec, i.e. total pixels / time taken

Total number of pixels in $m \times n$ image (array) = $m \times n$ pixels

Time taken for floating point operations = $(2 \times m \times n \times k \times k \text{ flops}) / 1638 \text{ Gflops/s}$

Time taken for memory operations = $(\text{sizeof arrayType}) \times (2 \times m \times n + k \times k) \text{ Bytes} / (136 \text{ Gbytes/s})$

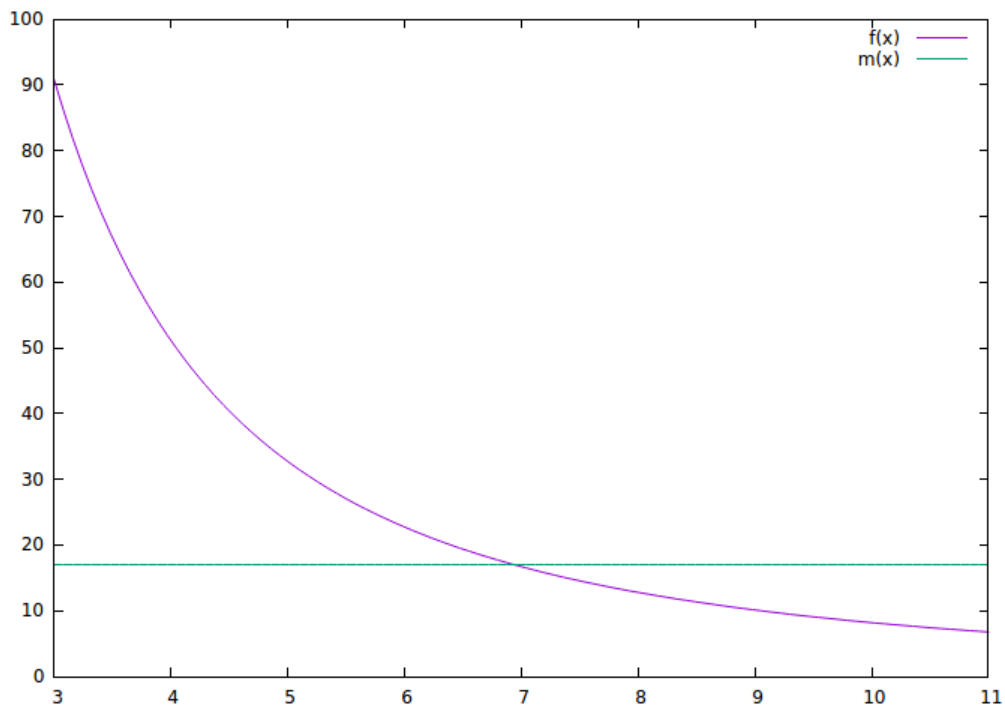
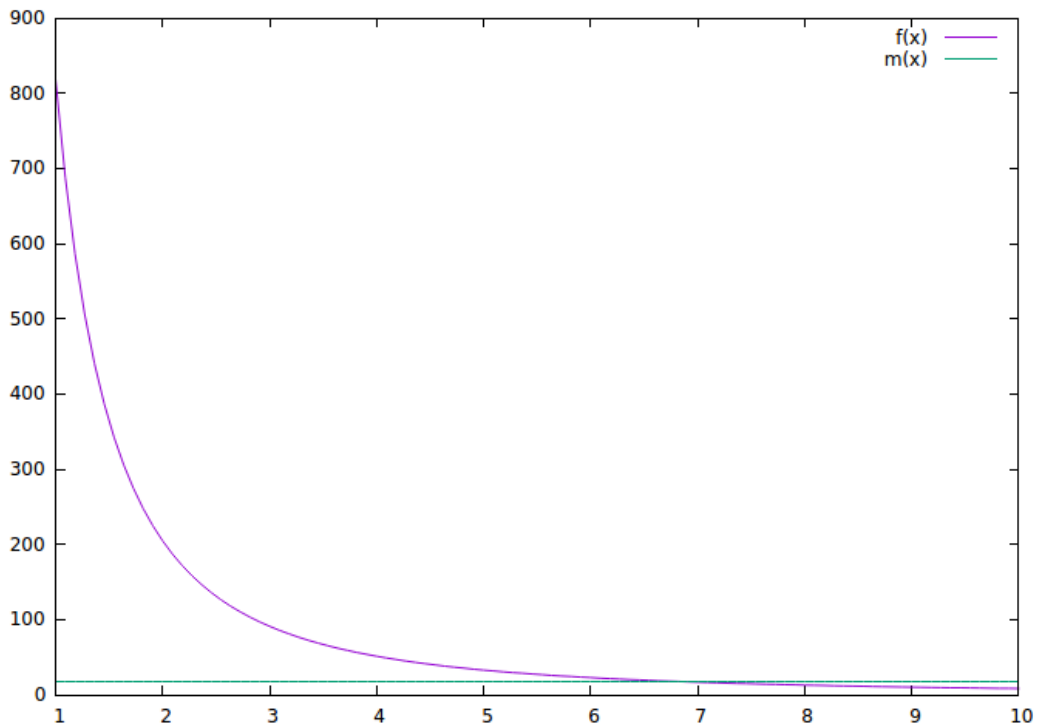
Performance for floating point operations = $(m \times n) \text{ pixels} / (2 \times m \times n \times k \times k \text{ flops}) / 1638 \text{ Gflops/s}$
= $819 / k^2 \text{ Gigapixels/s}$

Performance for floating point operations = $(m \times n) \text{ pixels} / 4 \times (2 \times m \times n + k \times k) / 136 \text{ Gflops/s}$
= $34 \times m \times n / (2 \times m \times n + k^2) \text{ Gigapixels/s}$

Considering $m=1024$ and $n=768$,

$f(x) = 819 / x^2 \text{ Gigapixels/s}$

$m(x) = 26738688 / (1572864 + x^2) \text{ Gigapixels/s}$



3) Basic Code:

```
#include<iostream>
#include <omp.h>
#include <immintrin.h>
#include<chrono>
#include<ctime>

using namespace std;
using namespace std::chrono;

int main(int argc, char **argv)
{
if(argc < 4)
{
    printf("Run parameters as follows:\nExecutable ImgHeight ImgWidth k\n");
}
else
{
    #pragma omp parallel
    {
        int ImageHeight=atoi(argv[1]), ImageWidth=atoi(argv[2]), k=atoi(argv[3]), sum=0;

        int **array=(int **) malloc(sizeof(int *) * ImageHeight);
        int **Earray=(int **) malloc(sizeof(int *) * ImageHeight);
        for(int r=0; r<ImageHeight; r++)
        {
            array[r] = (int *) malloc(sizeof(int) * ImageWidth);
            Earray[r] = (int *) malloc(sizeof(int) * ImageWidth);
        }
        int convArray[k][k];
        for(int i=0; i < ImageHeight; i++)
        {
            for(int j=0; j < ImageWidth; j++)
            {
                array[i][j]=1;
            }
        }

        for(int convR=0; convR < k; convR++)
        {
            for(int convC=0; convC < k; convC++)
            {
                convArray[convR][convC]=1;
            }
        }

        high_resolution_clock::time_point t1 = high_resolution_clock::now();
        for(int imgH=0; imgH < ImageHeight; imgH++)
        {
            for(int imgW=0; imgW < ImageWidth; imgW++)
            {
                for(int k1=0; k1 < k; k1++)
                {
                    for(int k2=0; k2 < k; k2++)
```

```

        {
            if((imgH+k1-1) >= 0 && (imgW+k2-1) >= 0 && (imgH+k1-1) < ImageHeight &&
(imgW+k2-1) < ImageWidth)
            {
                sum = sum + array[imgH+k1-1][imgW+k2-1];
            }
        }
    }
    Earray[imgH][imgW]=sum;
    sum=0;
}
}

high_resolution_clock::time_point t2 = high_resolution_clock::now();
duration<double> time_span = duration_cast<duration<double>>(t2 - t1);

printf(" Time taken for operations of array[%d][%d] with k as %d : %lf\n",ImageHeight, ImageWidth,
time_span.count(), k);

float fma=819/(k*k);
float Mem=(34*ImageHeight*ImageWidth)/((2*ImageHeight*ImageWidth) + (k*k));
printf("Performance in GigaPixels/s by FMA: Expected= %f\n",fma);
printf("Performance in GigaPixels/s by Memory: Expected= %f\n",Mem);
float expected=fma>Mem?fma:Mem;
printf("From above expected max performance= %f, actual= %f\n", expected, ImageHeight*ImageWidth/
(1000000*time_span.count()));

}
}
return 0;
}

```

Results:

For 35 Combinations:

Time taken for operations of array[1024][768] with k as 3 : 0.029402
Performance in GigaPixels/s by FMA: Expected= 91.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 91.000000, actual= 26.747960
Time taken for operations of array[1024][768] with k as 5 : 0.074796
Performance in GigaPixels/s by FMA: Expected= 32.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 32.000000, actual= 10.514345
Time taken for operations of array[1024][768] with k as 7 : 0.142653
Performance in GigaPixels/s by FMA: Expected= 16.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 5.512891
Time taken for operations of array[1024][768] with k as 9 : 0.232820
Performance in GigaPixels/s by FMA: Expected= 10.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 3.377858
Time taken for operations of array[1024][768] with k as 11 : 0.352814
Performance in GigaPixels/s by FMA: Expected= 6.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 2.229025
Time taken for operations of array[1024][768] with k as 13 : 0.487489
Performance in GigaPixels/s by FMA: Expected= 4.000000

Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 1.613232
Time taken for operations of array[1024][768] with k as 15 : 0.644381
Performance in GigaPixels/s by FMA: Expected= 3.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 1.220445
Time taken for operations of array[2048][2048] with k as 3 : 0.156685
Performance in GigaPixels/s by FMA: Expected= 91.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 91.000000, actual= 26.768992
Time taken for operations of array[2048][2048] with k as 5 : 0.399071
Performance in GigaPixels/s by FMA: Expected= 32.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 32.000000, actual= 10.510158
Time taken for operations of array[2048][2048] with k as 7 : 0.762319
Performance in GigaPixels/s by FMA: Expected= 16.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 5.502032
Time taken for operations of array[2048][2048] with k as 9 : 1.258840
Performance in GigaPixels/s by FMA: Expected= 10.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 3.331879
Time taken for operations of array[2048][2048] with k as 11 : 1.914493
Performance in GigaPixels/s by FMA: Expected= 6.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 1.584103
Time taken for operations of array[2048][2048] with k as 15 : 3.500098
Performance in GigaPixels/s by FMA: Expected= 3.000000
Performance in GigaPixels/s by Memory: Expected= 16.000000
From above expected max performance= 16.000000, actual= 1.198339
Time taken for operations of array[8192][8192] with k as 3 : 2.560644
Performance in GigaPixels/s by FMA: Expected= 91.000000
Performance in GigaPixels/s by Memory: Expected= -14.000000
From above expected max performance= 91.000000, actual= 26.207810
Time taken for operations of array[8192][8192] with k as 5 : 6.444124
Performance in GigaPixels/s by FMA: Expected= 32.000000
Performance in GigaPixels/s by Memory: Expected= -14.000000
From above expected max performance= 32.000000, actual= 10.413962
Time taken for operations of array[8192][8192] with k as 7 : 12.298233
Performance in GigaPixels/s by FMA: Expected= 16.000000
Performance in GigaPixels/s by Memory: Expected= -14.000000
From above expected max performance= 16.000000, actual= 5.456789
Time taken for operations of array[8192][8192] with k as 9 : 20.745547
Performance in GigaPixels/s by FMA: Expected= 10.000000
Performance in GigaPixels/s by Memory: Expected= -14.000000
From above expected max performance= 10.000000, actual= 3.234856
Time taken for operations of array[8192][8192] with k as 11 : 31.779112
Performance in GigaPixels/s by FMA: Expected= 4.000000
Performance in GigaPixels/s by Memory: Expected= -14.000000
From above expected max performance= 4.000000, actual= 1.532404
Time taken for operations of array[8192][8192] with k as 15 : 57.712156
Performance in GigaPixels/s by FMA: Expected= 3.000000
Performance in GigaPixels/s by Memory: Expected= -14.000000
From above expected max performance= 3.000000, actual= 1.162820
Time taken for operations of array[4194304][768] with k as 3 : 122.227446
Performance in GigaPixels/s by FMA: Expected= 91.000000

Performance in GigaPixels/s by Memory: Expected= 1.000000
From above expected max performance= 91.000000, actual= -8.784785
Time taken for operations of array[4194304][768] with k as 5 : 308.297330
Performance in GigaPixels/s by FMA: Expected= 32.000000
Performance in GigaPixels/s by Memory: Expected= 1.000000
From above expected max performance= 32.000000, actual= -3.482813
Time taken for operations of array[4194304][768] with k as 7 : 587.499734
Performance in GigaPixels/s by FMA: Expected= 16.000000
Performance in GigaPixels/s by Memory: Expected= 1.000000
From above expected max performance= 16.000000, actual= -1.827646
Time taken for operations of array[4194304][768] with k as 9 : 957.972798
Performance in GigaPixels/s by FMA: Expected= 10.000000
Performance in GigaPixels/s by Memory: Expected= 1.000000
From above expected max performance= 10.000000, actual= -1.120848
Time taken for operations of array[4194304][768] with k as 11 : 1456.733517
Performance in GigaPixels/s by FMA: Expected= 6.000000
Performance in GigaPixels/s by Memory: Expected= 1.000000
From above expected max performance= 6.000000, actual= -0.737089
Time taken for operations of array[4194304][768] with k as 13 : 2040.725037
Performance in GigaPixels/s by FMA: Expected= 4.000000
Performance in GigaPixels/s by Memory: Expected= 1.000000
From above expected max performance= 4.000000, actual= -0.526157
Time taken for operations of array[4194304][768] with k as 15 : 2652.391407
Performance in GigaPixels/s by FMA: Expected= 3.000000
Performance in GigaPixels/s by Memory: Expected= 1.000000
From above expected max performance= 3.000000, actual= -0.404820