```
void matrix_mult_nonthreaded(){
        int i,j,k;
        for(i=0;i<A_HEIGHT;i++){
                for(j=0;j<B_WIDTH;j++){
                        for(k=0;k<AB_SHARED;k++){
                                C[i][j]+=A[i][k]*B[k][i];
                        }
                }
        }
        return;
}

void* matrix_mult_threaded(void* id){
        int threadID=(int)id;
        int i,j,k;
        int num_rows_per_thread=A_HEIGHT/threads_available;
        int leftover_rows=A_HEIGHT%threads_available;
        int start_row=threadID*num_rows_per_thread;
        int stop_row=start_row+num_rows_per_thread;
        if(threadID==threads_available-1)
                stop_row=stop_row+leftover_rows;
        for(i=start_row;i<stop_row;i++){
                for(j=0;j<B_WIDTH;j++){
                        for(k=0;k<AB_SHARED;k++){
                                C[i][j]+=A[i][k]*B[k][i];
                        }
                }
        }
        return NULL;
}
```