1. (35 points) Problem 5.2. Run the stride-testing program on two different machines, preferably ones you can be the sole user on. To answer the paging questions you will need to use a much larger array. *Be sure to greatly reduce the number of iterations when testing paging performance.* Data sampling should not take very long. Be sure to explain your reading of the data (neatly annotate your graph), and include as much description of the machines used as possible. Discuss the differences between the two machines (you may just annotate the first graph and discuss the differences rather than annotating both and discussing the differences).

I’d like a few people who feel they have interesting graphs to make transparencies and present their memory system – let’s see how access times compare!

Here is a draft memtest.c program for use with gnuplot that you may wish to modify with other cache parameters or plotting programs.

2. (20 points) Consider a memory system with the characteristics:

<table>
<thead>
<tr>
<th></th>
<th>TLB</th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>16</td>
<td>32K</td>
<td>256K</td>
</tr>
<tr>
<td>Block Size</td>
<td>8KB</td>
<td>64B</td>
<td>64B</td>
</tr>
<tr>
<td>Associativity</td>
<td>16</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Miss Penalty</td>
<td>200</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>Hit Time</td>
<td>0*</td>
<td>0*</td>
<td>40</td>
</tr>
</tbody>
</table>

*through the wonders of pipelining*

And the following two indexing schemes for matrix multiply:

```c
EightByteDouble x[1024][1024];
EightByteDouble y[1024][1024];
EightByteDouble z[1024][1024];

int i, j, k;
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        for (k = 0; k < n; k++)
            x[i][j] += y[i][k] * z[k][j];
```

```c
EightByteDouble x[1024][1024];
EightByteDouble y[1024][1024];
EightByteDouble z[1024][1024];

int i, j, k;
for (i = 0; i < n; i++)
    for (k = 0; k < n; k++)
        for (j = 0; j < n; j++)
            x[i][j] += y[i][k] * z[k][j];
```

(a) Estimate the total data memory access time for the left code. Describe clearly (typset is nice!) how you arrived at that conclusion.

(b) Estimate the total data memory access time for the right code. Describe clearly (typset is nice!) how you arrived at that conclusion.

(c) Write code for the same loop working on 32x32 blocks. What will the total data access time be now?