Question # 1 [50 Points]

```c
int a[1024][512];
int b[1024][512];

for(i = 0; i < 1023; i++)
    for(j = 0; j < 511; j++)
        a[i][j] = b[i][j];
```

Assume a and b are allocated at address 0x00100000 and 0x00200000, respectively.

Assume the processor has a DMA engine available. The following 32-bit registers are used to setup DMA operation:

Register 1  Channel Control Register
Bit 0  Start DMA
Bit 1  DMA Completed Status (read only)
Bit 2  Enable Interrupts on Completion
Bit 3-31  Reserved

Register 2  Channel Source Address
Register 3  Channel Destination Address
Register 4  Count for 1st Dimension
Register 5  Count for 2nd Dimension

Program the DMA engine to achieve the for() loop functionality [50 Points]

<table>
<thead>
<tr>
<th>Register</th>
<th>Value in HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x5</td>
</tr>
<tr>
<td>2</td>
<td>0x00200000</td>
</tr>
<tr>
<td>3</td>
<td>0x00100000</td>
</tr>
<tr>
<td>4</td>
<td>0x1000</td>
</tr>
<tr>
<td>5</td>
<td>0x200</td>
</tr>
</tbody>
</table>
Question # 2

Assume a signal called SIGDMA is generated on a Linux machine upon DMA completion. After starting the DMA engine, assume the program will wait while a global variable dmaBusy is equal to 1. Complete the following C program. [50 Points]

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

int dmaBusy;

void dmaISR()
{
    dmaBusy = 0;
}

main()
{
    signal(SIGDMA, dmaISR);
    setupDMA(); /* This function is used to initialize and start the DMA engine */

    while(dmaBusy == 1)
    {
        // do nothing
    }
}
```