

Tutorial 5: Memory Access

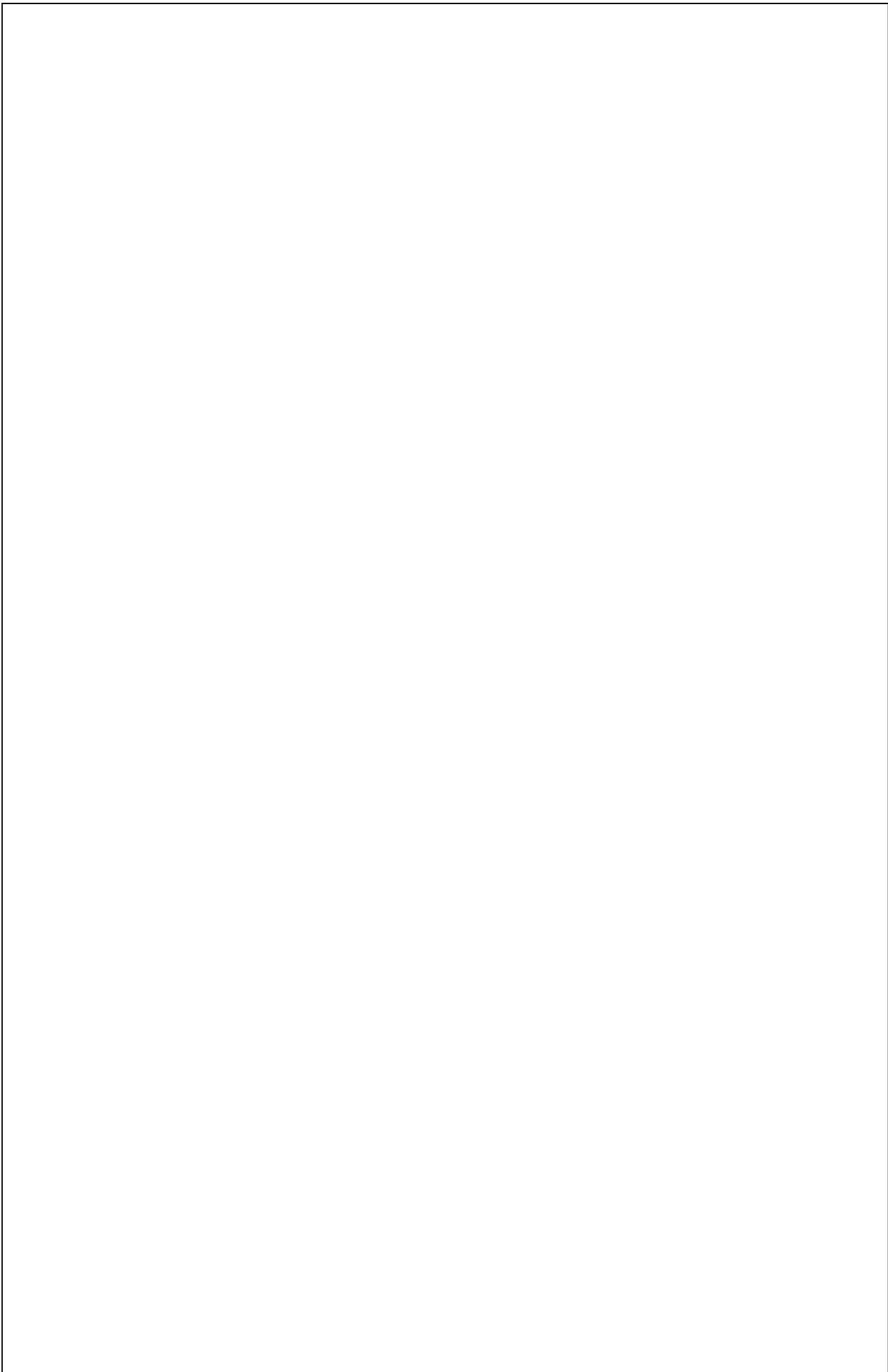
Problem 1: Registers, Memory and the Code Compilation

Consider the C code in Figure 1. Draw a memory map for all the variables in the program.

```
int main (void)
{
int a[ ] = {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
int b[ ] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19};
int i,temp;
    for(i=0;i<10;i=i+1)
    {
        temp = b[i];
        b[i] = a[i];
        a[i] = temp;
    }
return 0;
}
```

Figure 1: Program Data Structures





Problem 2: Little and Big Endian Machines

Consider the C code in Figure 4.

What are the outputs of the printf statements?

Write an equivalent assembly language program for the part corresponding to (c, *(c+1), *(c+2), *(c+3))

```
#include <stdio.h>

int main (void)
{
    int a=287454020;
    char *c;
    c = ( char *) &a;
    printf("Number as integer = \"%x\"\n\n", a);
    printf("Number as 4 characters = \"%x%x%x%x\"\n\n", *c, *(c+1), *(c+2),
*(c+3));
    return 0;
}
```

Figure 4: Assembly Program on Big and Little Endian

Problem 3: Word Alignment

Consider the C code in Figure 7.

What are the outputs of the printf statements?

Write an equivalent assembly language program for the parts corresponding to (a1, a2, a3, a4, a5) and (*c).

```
#include <stdio.h>

int main (void)
{
    static char a1=20, a2 =54, a3 = 74, a4 = 28, a5 = 34;
    int *c;
    c = ( int *) &a2;
    printf("Number as 5 characters = \"%x%x%x%x%x\"\\n\\n",a1, a2, a3, a4, a5);
    printf("Number as integer = \"%x\"\\n\\n", *c);
    return 0;
}
```

Figure 7: The Outputs of printf Statements

