#1.

a) Write the appropriate save instruction to allocate stack space for the following local variables and any padding.

```c
short a;
char b;
int c;
char d;
unsigned short e;
unsigned long f;
```

```
save _________ , ______________________________  , _________
```

(Use the formula, not an absolute value)

b) Write the appropriate unoptimized SPARC assembly instructions using the above local variables.

```assembly
f = 8675309;
```

```assembly
a = 0xBABE;
```

```assembly
d = ’A’;
```

```assembly
c = e;
```

(OVER)
#2.  

**a)** Write the appropriate **save** instruction to allocate stack space for the following local variable declaration.

```c
char a[6];
```

**save** _________ , ______________________________ , _________

(Use the formula, not an absolute value)

**b)** Write the appropriate instructions to perform the following assignment statements.

```c
a[1] = a[3];
```

________________________

________________________

```c
a[2] = a[5];
```

________________________

________________________

```c
char *ptr; /* ptr mapped to %l2 */
```

```c
ptr = &a[0];
```

________________________

________________________

```c
ptr++; /* ptr mapped to %l2 */
```

________________________

________________________

```c
char x = *ptr; /* x mapped to %l0; ptr to %l2 */
```

________________________

________________________

```c
*ptr = x; /* x mapped to %l0; ptr to %l2 */
```

________________________

________________________

#3. Write the equivalent C expression the compiler really uses for an array index access.

```c
int a[5];
```

a[i] is equivalent to _____________________________ in C.