#1.

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

\[
\begin{align*}
\text{unsigned long} & \quad a; \\
\text{short} & \quad b; \\
\text{char} & \quad c; \\
\text{int} & \quad d; \\
\text{char} & \quad e; \\
\text{unsigned short} & \quad f;
\end{align*}
\]

\[
\text{save } \underline{\text{_______}}, \underline{\text{_______}}, \underline{\text{_______}}.
\]

(Use the formula, not an absolute value)

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

\[
e = c;
\]

\[
f = 0x\text{BABE};
\]

\[
d = 8675309;
\]

\[
a = b;
\]
#2.

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

```plaintext
double a[6];
```

```plaintext
save __________ , ______________________________ , __________
(Use the formula, not an absolute value)
```

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

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

```plaintext
________________________
________________________
```

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

```plaintext
________________________
________________________
```

```plaintext
double *ptr; /* ptr mapped to %l2 */
ptr = &a[0];
```

```plaintext
________________________
```

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

```plaintext
________________________
```

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

```plaintext
________________________
```

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

```plaintext
________________________
```

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

```plaintext
short a[10];
```

```plaintext
a[i] is equivalent to _____________________________ in C.
```