#1.

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

\begin{verbatim}
long    a;
char    b;
unsigned short  c;
int     d;
char    e;
short   f;
\end{verbatim}

\begin{verbatim}
save   ,  ______________________________  ,  _______
\end{verbatim}

(Use the formula, not an absolute value)

b) Write the appropriate \texttt{unoptimized} SPARC assembly instructions using the above local variables.

\begin{verbatim}
b = c;
\end{verbatim}

\begin{verbatim}
f = 0xCAFE;
\end{verbatim}

\begin{verbatim}
d = -8675309;
\end{verbatim}

\begin{verbatim}
a = e;
\end{verbatim}
#2. a) Write the appropriate `save` instruction to allocate stack space for the following local variable declaration.

```c
double a[5];
```

`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
________________________
________________________
```

```c
a[2] = a[4];
```

```c
________________________
________________________
```

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

```c
________________________
```

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

```c
________________________
```

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

```c
________________________
```

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

```c
________________________
```

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

```c
short a[10];
```

`a[i]` is equivalent to `______________________________` in C.