#1.

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

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

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

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

```
d = b;
```

```
a = -9876;
```

```
e = 'B';
```

```
a = c;
```

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

```
short a[9];
```

```
save _______ , ______________________________ , _________
(Use the formula, not an absolute value)
```

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

```
a[7] = a[4];
________________________
________________________

a[5] = a[1];
________________________
________________________
```

```
short *ptr; /* ptr mapped to %l5 */
ptr = &a[1];
________________________
```

```
++ptr; /* ptr mapped to %l5 */
________________________
```

```
short d = *ptr; /* d mapped to %l2; ptr to %l5 */
________________________
```

```
*ptr = d; /* d mapped to %l2; ptr to %l5 */
________________________
```

#3. Write the equivalent C expression for an array index access. Remember scaling is done automatically by the compiler.

```
float a[10];
```

```
a[i] is equivalent to _____________________________ (equivalent pointer expression).
```