#1.
a) Write the appropriate save instruction to allocate stack space for the following local variables and any padding.

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

```
save _______ , ______________________________  , _________
(Use the formula, not an absolute value)
```
b) Write the appropriate unoptimized SPARC assembly instructions using the above local variables.

```
f = b;
```

```
d = 9876;
```

```
e = ’B’;
```

```
c = a;
```

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

```c
float a[9];
```

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

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

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

```
________________________
________________________
```

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

```
________________________
________________________
```

```c
float *ptr; /* ptr mapped to %l5 */
ptr = &a[1];
```

```
________________________
________________________
```

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

```
________________________
________________________
```

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

```
________________________
________________________
```

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

```
________________________
________________________
```

#3. Write the equivalent C expression how the compiler really uses the name of an array.

```c
double a[10];
```

```
a is equivalent to ____________________________ in C.
```