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

```
char a;
long b;
short c;
int d;
```

```
save __________ , ______________________________  , __________
```

b) Write the appropriate instructions to store the value `\n` in local variable `a` as defined above (hint: you cannot assign an immediate constant value directly to memory — you can only store from a register to memory).

```
________________________ ! \n -> %l0
________________________ ! %l0 -> local variable a
```

c) Write the appropriate instructions to store the value `0x87654321` in local variable `b` as defined above (same hint as above).

```
________________________ ! 0x87654321 -> %l0
________________________ ! %l0 -> local variable b
```

d) Write the appropriate instruction to place the value in local variable `c` into register `%l0`.

```
________________________ ! c -> %l0
```

e) Write the appropriate instruction to place the value in local variable `a` into register `%l0`.

```
________________________ ! a -> %l0
```

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

```
short a[8];
```

```
save  ________ ,  ______________________________  ,  ________
```

b) Write the appropriate instructions to perform the following assignment statements, assuming variable \( i \) is mapped to \( %l0 \) and \( ptr \) is mapped to \( %l1 \).

```
short i; /* i mapped to %l0 */
i = a[2];

________________________ ! a[2] -> %l0

i = a[5];

________________________ ! a[5] -> %l0

short *ptr; /* ptr mapped to %l1 */
ptr = a;

________________________ ! a -> %l1

++ptr; /* ptr mapped to %l1 */

________________________ ! ++ptr

i = *ptr; /* i mapped to %l0; ptr to %l1 */

________________________ ! *ptr -> %l0

*ptr = i; /* i mapped to %l0; ptr to %l1 */

________________________ ! %l0 -> *ptr