A. Questions

1. Suppose a, b and c are integral type variables that have been assigned the values `a = 12`, `b = 5` and `c = -6`. Determine the value of each of the following arithmetic expressions.

```
a) a + b
b) 2 * b + 3 * ( a - c )
c) a / b
d) a % b
e) a * b / c
f) a * (b / c)
```

> Answers:
> a) 17
> b) 64
> c) 2 (Integer division, fractional part is truncated)
> d) 2 (Modulo operation, returns the remainder)
> e) -10
> f) 0

2) A C program contains the following declarations:

```
int i, j;
long ix ;
short s;
float x;
double dx;
char c;
```

Determine the data type of each of the following expressions.

```
a) i + c
b) dx + ix
c) ((int) dx) + ix
d) i + x
e) ix + j
```

> Answers: (see conversion hierarchy)
> a) int
> b) double
> c) ~~int~~ long
> d) float
> e) long

3) Determine the data type of each of the following literals.

```
a) 150
b) 15.3
c) '2'
```

d) `3.14f`
e) `0x7fffU`
f) `01234L`

Answers:
a) int
b) double
c) char
d) float
e) unsigned int
f) long

4) Declare an unsigned long integer constant (using `const` keyword) with identifier `ulconst` and value 127,745.

Best answer:
```
const unsigned long ulconst = 127745LU; //or 127745lu
```

But this will also work due to implicit conversion:
```
const unsigned long ulconst = 127745;
```

5) Declare an unsigned long integer constant (using macro) with identifier `ULCONST` and value 127,745.

Best answer:
```
#define ULCONST 127745LU //or 127745lu
```

This will work but may generate warnings depending on how `ULCONST` is used:
```
#define ULCONST 127745
```

6) Declare an unsigned integer constant (using macro) with with identifier `STATUS` and value 48,879 in hexadecimal.

Answer:
```
#define STATUS 0XBEEF // or 0xbeef
```

7) Declare an character constant (using `const` keyword) with with identifier `SECRET_KEY` and the newline character as value.

Answer:
```
const char SECRET_KEY = '\n';
```