Example 1: The first line would select out the 1st, 4th, 7th rows and the 2nd through 5th column of `X`.

The second line would select everything except the 6th, 7th, and 8th columns of `X`.

The third line would only select out the rows where the 7th column of that row is greater than 1 in absolute value.

Example 2: One way to do this would be

```
n <- 0
S <- 0
while( S < 50 )
{
    n <- n + 1
    S <- S + (n^(3/2))
}
print(n)
[1] 7
```

Example 3: This program investigates the sampling distribution of $\overline{X}$ from a sample of size $n = 20$ from an exponential(1/3) distribution. Since the sample mean has expected value equal to $1/\lambda$, we would expect that `mean(M)` would be around 3. Also, $\text{var}(\overline{X}) = \text{var}(X)/20$ so we would expect `var(M)` to be around $(1/\lambda^2)/20 = 9/20 = .45$.

So the second answer is correct.

Example 4: This program generates $N(0, 1)$ random variables, then selects out those that are between 0 and 2, and then takes the sample mean of the remaining values. So, this estimates

$$E(X|0 \leq X \leq 2)$$

where $X \sim N(0, 1)$.

Example 5: One way to do this is

```
X <- rnorm(10000)
w <- which(X < 0)
X[w] <- 0
X[-w] <- sqrt(X[-w])
```

Example 6: Rejection sampling accepts a candidate if

$$U \leq \frac{p(X)}{M \cdot g(X)}$$

Plugging in the provided values for X, and M, the right hand side of this inequality evaluates to

$$\frac{p(1.36)}{1.52 \cdot g(1.36)} \approx 0.932$$

which is indeed greater than $U = .632$, so we would accept this draw.

Example 7: The inversion method requires you to invert the CDF, so the first step is to calculate the CDF:

$$F(x) = \int_1^x \lambda y^{-\lambda-1} dy = -\frac{1}{y^\lambda}\bigg|_1^x = 1 - \frac{1}{x^\lambda}$$

It is straightforward to see that

$$F^{-1}(u) = \left(\frac{1}{1-u}\right)^{1/\lambda}$$

The inversion method works by plugging Uniform$(0,1)$'s into the inverse CDF, so a one line function to generate n samples from this distribution would be

```
rF <- function(n, L) (1/(1 - runif(n)))^(1/L)
```

To estimate $E(X^{2.736})$ when $\lambda = 3$ based on $n = 1000$ samples you could type

```
mean(rF(1000, 3)^2.736)
```

Example 8: For the first part, this can be viewed as

$$E(10 \cdot (U-5)^2)$$

where U is Uniform(-3,3), so the code using k monte carlo samples is

```
f <- function(x) 6*exp(-abs(x))
X <- runif(k, -3, 3)
mean( f(X) )
```

using the $N(0,1)$ distribution for rejection sampling, this is

```
f <- function(x) 6*exp(-abs(x))
w <- function(x) dunif(x,-3,3)/dnorm(x)
X <- rnorm(k)
mean( w(X)*f(X) )
```

The importance sampling is expected to perform better since the integrand will match the shape of the density being integrated against more closely.

Example 9: The first step with Newton-Raphson is to calculate the derivatives of the objective function, $f(x) = \exp(-x^2 + 3x - 4)$. These are

$$f'(x) = (-2x+3)f(x)$$

and

$$f''(x) = -2f(x) + (-2x+3)f'(x)$$

So, the first iteration will take you to

$$1 - f'(1)/f''(1)$$

Evaluating directly,

$$f'(1) = (-2 \cdot 1 + 3) f(1) = f(1)$$

and

$$f''(1) = -2 \cdot f(1) + (-2 \cdot 1 + 3) f'(1) = -2 \cdot f(1) + f(1) = -f(1)$$

so $f'(1)/f''(1) = -1$ and

$$x_1 = 1 - (-1) = 2$$

is where you will be after 1 iteration.