1.

- 
```
void sum(double A[][M], double B[][M], double C[][M], double D[][M])
{
  int i,j;

  for (i=0; i<N; i++)
    {
      for (j=0; j<M; j++)
        {
          C[i][j] = A[i][j] + B[i][j];
          D[i][j] = A[i][j] - B[i][j];
        }
    }
}
```

- 
```
double fofx( double x )
{
  return( (sqrt(x+1)*pow(x+1,2*x))/(2*cos(x)) );
}
```

- 
```
double meanval(int n, double x[])
{
  int i;
  double mean = 0.0;

  for (i=0; i<n; i++)
    {
      mean += x[i];
    }
  mean /= n;
  return(mean);
}
```

2. Main points

- draw the picture as given in the notes
- explain (briefly) the proceedure as represented in the picture (2-3 sentances)
- method arises from a Taylor series expansion ie. $f(x+\delta) = f(x) + \delta f'(x) + \ldots$
- restricting to the first 2 terms $\Rightarrow \delta = -f(x)/f'(x)$
- signal for a root at $x$ is $\delta = 0$ and $f(x) = 0$.
- Newton Raphson algorithm is $x_{n+1} = x_n + \delta$

The proof of quadratic convergence is in the notes (referred to as a recurrence relation).
The 2 roots are $x = 4.236$ and $x = -0.236$ (you should show workings via Newton Raphson for these results).

3. The sorting program for this question is on-line at `http://www.maths.tcd.ie/ ryan/teaching/sele` The only change you need to make is to read the numbers from the keyboard (using a `scanf` statement) rather than writing them into the program.

4. (a) $(((x/y) * i) + j) \rightarrow 1.6666$
(b) $(((x < y)\&\&(y! = 0.0))||(i == j)) \rightarrow$ TRUE(= 1)
(c) $(((!x) * y)/2.0) \rightarrow 0$

**Primary difference between an `array` and a `structure`?**
Array members must be all of the same type. A structure can have members of different types.

Describe the memory map:

| Byte Address | Value | Variable Name |
|---|---|---|
| 28-32 | | employees[1].age |
| 20-28 | | employees[1].salary |
| 16-20 | | employees[0].age |
| 8-16 | | employees[0].salary |
| 4-8 | | p |
| 0-4 | | i |

Trace what happens during execution :

| Byte Address | Value | Variable Name |
|---|---|---|
| 28-32 | 27 | employees[1].age |
| 20-28 | 22000.00 $\longrightarrow$ 24200.00 | employees[1].salary |
| 16-20 | 25 | employees[0].age |
| 8-16 | 20000.00 $\longrightarrow$ 22000.00 | employees[0].salary |
| 4-8 | $8_{mem} \longrightarrow 20_{mem}$ | p |
| 0-4 | $0 \longrightarrow 1 \longrightarrow 2$ | i |

(*Note that these two memory maps can be combined in one for your exam answer - I split them up to show you the two distinct parts*).

5. $(223)_{10} = (df)_{16} = (11011111)_2$. The `HEX` pattern for this is (using 2's complement)

$$ffffff21$$

$(fab)_{16} = (111110101011)_2 = (4011)_{10}$. The `HEX` pattern for this is (using 2's complement)

$$fffff055$$

$(1.714)_{10} = (1.b6)_{16} = (1.10110110)_2$. The `HEX` pattern for this is (using IEEE)

$$3fdb0000$$

$(10001.01011)_2 = (11.58)_{16} = (17.34375)_{10}$. The `HEX` pattern for this is (using IEEE)

$$418ac000$$

1megabyte = 1048576 bytes (1024 * 1024 ie $(2^{10} * 2^{10})$ bytes. This is 8388606 bits.

Answer = 8388606 + 8388606/8 = 9437184 bits = 1.125Mb.

6. The 2 coupled equations are

$$\begin{aligned} \frac{dy}{dt} &= z(t) \\ \frac{dz}{dt} &= 1 - 2z(t) \end{aligned}$$

Define the Euler algoritm: to solve the equation $\frac{dy}{dx} = f(x, y)$, the Euler algorithm is

$$\begin{aligned} y_0 &= A \\ y_{n+1} &= y_n + hf(x_n, y_n) \end{aligned}$$

For this problem:
$t_1$ ie $t = 0.1$

$$\begin{aligned} y_1 &= y_0 + hz_0 = -1.0 \\ z_1 &= z_0 + h(1 - 2z_0) = 0.1 \end{aligned}$$

$t_2$ ie $t = 0.2$

$$\begin{aligned} y_2 &= y_1 + hz_1 = -0.99 \\ z_2 &= z_1 + h(1 - 2z_1) = 0.18 \end{aligned}$$

$t_3$ ie $t = 0.3$

$$\begin{aligned} y_3 &= y_2 + hz_2 = -0.972 \\ z_3 &= z_2 + h(1 - 2z_2) = 0.244 \end{aligned}$$

$t_4$ ie $t = 0.4$

$$y_4 = y_3 + hz_3 = -0.9476$$

Answer: $y = -0.9476$ at $t = 0.4$.

7. The derivation of Simpson's rule is in the notes (look for the picture of the parabola!). The extended rule is written in the notes and the derivation of the error is also there.

   You should have derived

   $$\text{error term } = \frac{1}{180}\frac{(b-a)^5}{n^4} max(f'''' \Big|_a^b)$$

   Then for $f(x) = \dfrac{1}{x^4}$, $max(f'''')_a^b = max\left(\dfrac{840}{x^8}\right)_{x=1}^{x=4} = 840.$

   Therefore

   $$\begin{aligned} \frac{1}{180}\frac{(4-1)^5}{n^4} 840 &< 10^{-4} \\ n &> 51 \end{aligned}$$

   Therefore $n = 52$ sub-intervals are needed.

8. The main points are

   - Call by value pass values of variables to functions. Call by reference uses pointers to pass the address of variables in `main` to the function
   - Call by value results in a local copy in the function, call by reference does not result in a local copy
   - To return a result, or change the value of a variable in the `main` program requires a `return` statement when using call by reference. This can be accomplished without a `return` statement using call by reference.

   ```
   a daft program
   XYZ VTYZ
   she.
   ells.
   ea.
   hells.
   y.
   he.
   eashore
   ```