Show all your work and indicate your reasoning in order to receive the most credit.

Name:

1. (3pt) Convert decimal number 137 to binary form.
2. (3pt) Convert binary number 1110011 to decimal form.
3. (4pt) How many CPUs are there in the laptop used by the instructor in class? How much RAM?
4. (80pt) Write a C program that implements the algorithm presented below.

   5. An algorithm for π. We now present the details of our exponentially converging algorithm for calculating the digits of π. Twenty iterations will provide over two million digits. Each iteration requires about ten operations. The algorithm is very stable with all the operations being performed on numbers between ½ and 7. The eighth iteration, for example, gives π correctly to 694 digits.

   THEOREM 2. *Consider the three-term iteration with initial values*

   $$\alpha_0 := \sqrt{2}, \quad \beta_0 := 0, \quad \pi_0 := 2 + \sqrt{2}$$

   given by

   (i) $\alpha_{n+1} := \frac{1}{2}(\alpha_n^{1/2} + \alpha_n^{-1/2})$,

   (ii) $\beta_{n+1} := \alpha_n^{1/2}\left(\frac{\beta_n + 1}{\beta_n + \alpha_n}\right)$,

   (iii) $\pi_{n+1} := \pi_n \beta_{n+1}\left(\frac{1 + \alpha_{n+1}}{1 + \beta_{n+1}}\right)$.

   Then π_n converges exponentially to π and

   $$|\pi_n - \pi| \leqq \frac{1}{10^{2^n}}.$$

   Print the result of each iteration of the algorithm. Make as many iterations as necessary to achieve the accuracy better than 10^{-10}. Provide printouts of your (formatted) C code, your makefile, and (nicely formatted) output of your program.

5. (5pt) An adaptive integration routine (a) uses Simpson's rule as the basic algorithm and (b) always subdivides the interval into equal halves. The routine was used to calculate the following integral:

   $$\int_0^{\pi} \sin^2(64x)\, dx.$$

   What was the result of the integration? What is the correct result?

6. (5pt) As a part of the solution of a particular problem you need (repeated) calculations of the following expression:

$$\frac{1}{1-\sqrt{1-x}}. \tag{1}$$

for small x, $x \sim \epsilon$, where ϵ is machine epsilon.

What troubles you expect when using Eq. (1)? Rewrite Eq. (1) to avoid those troubles.