I don't think a sufficiently clever algorithm could save us, because inherent in the problem is that the roots are incredibly sensitive to the coefficients. It's nothing wrong with our algorithm per se.

Problem 8

Part (d)

Below, I've written a code to demonstrate that the algorithm we have made for calculating the terms in this series is accurate.

```
In[ ]:= y = 0.; (* Our "guess" *)
n = 25; (* The higher value we start at. *)
k = 20; (* The desired value *)
For[i = n, i > k, i--,
 y = (E - y) / i;
]
yk = y;
Print[N[y]]

0.123804
```

```
In[ ]:= ykInt = NIntegrate[x^k Exp[x], {x, 0, 1}]
Out[ ]= 0.123804
```

At least by inspection, they appear the same, so it appears our algorithm has worked. Let's subtract the two to find what the relative error is, assuming that the integral is exact to machine precision.

```
In[ ]:= Log2[Abs[ykInt - yk]]
Out[ ]= -25.9144
```

The error is less than 2^{-25}, and our goal was only to get it to better than 2^{-23}, so I would say we have met and exceeded our tolerances. It helps to use reasonable upper bounds on everything all the time. Printing these side by side, we can visually see the slight difference.

```
In[ ]:= NumberForm[yk, 16]
NumberForm[ykInt, 16]
Out[ ]//NumberForm=
0.1238038465745379
Out[ ]//NumberForm=
0.1238038307625704
```

8)

Recurrance in reverse

a) $$\frac{e - y_{n+1}}{n+1} = y_n$$

we now ask, given y_{n+1}'s error, what's the error in y_n?

$$\Rightarrow \frac{e(1+\varepsilon) - y_{n+1}(1+\varepsilon_{n+1})}{n+1} = y_n$$

$$\frac{e - y_{n+1}}{n+1} + \frac{e\varepsilon - y_{n+1}\varepsilon_{n+1}}{n+1} = y_n(1+\varepsilon_n)$$

Since we intend it, it's fair to assume

$$\varepsilon_{n+1} y_{n+1} >> e\varepsilon \quad \text{or} \quad \frac{y_{n+1}}{e} >> \varepsilon/\varepsilon_{n+1}$$

$$y_n \varepsilon_n = \frac{e\varepsilon - y_{n+1}\varepsilon_{n+1}}{n+1}$$

$$\Rightarrow |\varepsilon_n| = \left|\frac{y_{n+1}}{y_n}\right|\left|\frac{\varepsilon_{n+1}}{n+1}\right|$$

Okay, then what's $|\varepsilon_{n-1}|$?

$$|\varepsilon_{n-1}| = \left|\frac{y_n}{y_{n-1}}\right|\left|\frac{\varepsilon_n}{n}\right|$$

$$= \left|\frac{y_{n+1}}{y_{n-1}}\right|\left|\frac{\varepsilon_{n+1}}{n(n+1)}\right|$$

ε_{n-2}?

$$|\varepsilon_{n-2}| = \left|\frac{y_{n-1}}{y_{n-2}}\right|\left|\frac{\varepsilon_{n-1}}{n-1}\right|$$

$$= \left|\frac{y_{n+1}}{y_{n-2}}\right|\left|\frac{\varepsilon_{n+1}}{(n-1)n(n+1)}\right|$$

The pattern seems to be that, for $k < N$

$$|\mathcal{E}_k| = \left|\frac{y_N}{y_k}\right|\left|\mathcal{E}_N \frac{k!}{N!}\right|$$

Let's prove this.

$$|\mathcal{E}_{k-1}| = \left|\frac{y_k}{y_{k-1}}\right|\left|\frac{\mathcal{E}_k}{k}\right|$$

$$= \left|\frac{y_N}{y_{k-1}}\right|\left|\mathcal{E}_N \frac{(k-1)!}{N!}\right|$$

It seems to hold. Now, $|y_N/y_k| < 1$, and if N is large, then we have approximately:

$$y_N = \int_0^1 dx\, x^N e^x \approx e^1 \int_0^1 dx\, x^N = \frac{e}{N+1}$$

The portion of the integral where $x < 1$ is basically zero, and so taking e^x to be equal to its value at 1 introduces little error. So the y_N asymptotically approach zero, and we even have a good initial guess for y_N.

Now, $g_k(y_N) = y_k$.

$$(\text{cond } g_k) = \frac{d\log g_k}{d\log y_N}$$

$$= \frac{\partial \log g_k}{\partial \log y_{k+1}} \times \frac{\partial \log y_{k+1}}{\partial \log y_{k+2}} \times \cdots \times \frac{\partial \log y_{N-1}}{\partial \log y_N}$$

Having reversed the recurrence relation, this is easy.

$$\frac{\partial y_n}{\partial y_{n+1}} = -\frac{1}{n+1} \Rightarrow \frac{\partial \log y_n}{\partial \log y_{n+1}} = \frac{-1}{n+1}\frac{y_{n+1}}{y_n}$$

So we see that:

$$\boxed{(\text{Cond } g_k) = (-1)^{N-k} \frac{y_N}{y_k} \frac{k!}{N!}}$$

We could have backed this out of our previous expression for $\varepsilon_k(\varepsilon_N)$.

b) Let $\varepsilon_k < \varepsilon$ be the required tolerance.

Let the initial $\varepsilon_N = |y_N|$, and so

$$\varepsilon_k = \left| \frac{y_N^2}{y_k} \frac{k!}{N!} \right|$$

Well, $y_k > y_N$, so taking $\frac{y_N^2}{y_k} \rightarrow y_N > \frac{y_N^2}{y_k}$

$$\varepsilon_k < \left| y_N \frac{k!}{N!} \right|$$

Also, $y_N \lesssim \frac{e}{N+1}$ (since $e^x x^N < x^N$ on $(0,1]$)

$$\Rightarrow \quad \varepsilon_k < \frac{e k!}{(N+1)!}$$

So to ensure machine precision, we need to start at whatever N satisfies, given our k,

$$\boxed{\varepsilon_k < \frac{e k!}{(N+1)!} < \varepsilon}$$

There's no inverse factorial, so it's hard to write an exact expression.

c) $\varepsilon = eps.$ $K = 20.$

$$\frac{e \times 20!}{(N+1)!} < eps = 2^{-23} \quad \text{(single precision)}$$

If we have to, we can use Stirling's approximation for $n!$

we would want to impose a lower bound on $n!$.

the bound is:

$$\varepsilon_k < \frac{eK!}{(N+1)!} < \frac{eK!}{\sqrt{2\pi}\, n^{n+1/2} e^{-n} e^{\frac{1}{12n+1}}} \quad \text{(Robbins 1955)} \quad (n = N+1)$$

$$\varepsilon_k < \frac{K!\, e^{N+2}}{\sqrt{2\pi}\,(N+1)^{N+3/2}}\, e^{\frac{-1}{12N+13}}$$

For $K=20$, what N makes this less than the machine epsilon? It's easiest to solve this in log space:

$$-23 \log 2 > \log(20!) + (N+2) - \frac{1}{12N+13} - (N+\tfrac{3}{2})\log(N+1) - \tfrac{1}{2}\log(2\pi)$$

(I numerically solved this)

Rounding up, we find that $\boxed{N = 25}$ Not actually so far from 20! Nice.

What about double precision? We just let $-23\log 2 \longrightarrow -52\log 2$

$\Rightarrow$ To double precision accuracy, we need to go from an $\boxed{N = 31}$