Standard

Gfortran

```
 Found          135
```

onlinegdb

```
 Found          135
```

Gfortran optimisation

-o

```
 Found          135
```

-o2

```
 Found          135
```

-o3

```
 Found          135
```

Commenting out line 14

```
Found          135 – no effect
```

-o

```
 Found          135
```

-o2

```
 Found          135
```

-o3

```
 Found          135
```

using integer,parameter :: my_kind=kind(1.d0)

Gfortran

```
 Found           82
```

onlinegdb

```
 Found           82
```

Gfortran optimisation

-o

```
 Found           82
```

-o2

```
 Found           82
```

-o3

```
 Found           82
```

Reduced number of failed inverses using using double precision as apossed to single precision, as with more bits avalible to represent floating point numbers the truncation errors that are inate in floating point numbers have less of an effect on the answer.

Code

```
program q3_ac2071
implicit none
integer, parameter :: precision=kind(1.0)
integer, parameter :: double=kind(1.d0)

real (kind=precision) :: a,b
real (kind=double) :: c_d,d_d

a=tiny(1.0)
b=huge(1.0)

c_d=tiny(1.d0)
d_d=huge(1.d0)

print*, "smallest single precision number:",a,"binary digits:", digits(a)
print*, "largest single precision number:",b,"binary digits:", digits(b)
print*, "snallest double precision number:",c_d,"binary digits:", digits(c_d)
print*, "largest double precision number:",d_d,"binary digits:", digits(d_d)

end program q3_ac2071
```