Yanes Garcia

Math 4670

Basic Programs

1. This Fortran program determines the largest integer that a system can represent. It is like the Fortran program which find the largest real value; however, the variable ‘big’ is represented like an integer in this case. The program consists in setting some integer variable to one and then keep doubling it until we find some odd behavior. Like I try to find an integer variable, my variable must not have numbers with decimal points as a variable of type real does. Code for this is:

```
implicit none
integer :: i
integer :: big
big=1
do i=1,050
big=big*2
print*,big
end do
stop
end
```

I’ll show only the relevant part of the output.

248
16
32
64
128
256
512

And so on and so forth. I am going to show the last numbers on the sequence which is created when I try to find the largest integer.

134217728
268435456
536870912
1073741824
-2147483648
00

The largest integer value that my system can represent is about 1073741824, which is the last coherent value that appears in my sequence, and its double. At first, the sequence is satisfactory because it increases like it should do it; however, the end is a mystery because we get a negative number and several zeros.

2. This Fortran program determines the smallest positive real value that my system can represent. It is like the Fortran program which find the largest real value; however, this program wants to find the smallest number. In this case, the program consists in setting some real variable to one and then keep dividing it by two until we find some odd behavior. Code for this follows:

```
implicit none
integer :: i
real :: small
small=1.0
do i=1,150
small=small/2.0
if(small<=0) then
stop
```

```
end if
print*,small
end do
stop
end
```

I'll show only the relevant part of the output.

0.500000

0.250000

0.125000

6.250000E-02

3.125000E-02

1.562500E-02

7.812500E-03

And so on and so forth. I am going to show the last numbers on the sequence which is created when I try to find the smallest positive real variable value.

1.880791E-37

9.403955E-38

4.701977E-38

2.350989E-38

1.175494E-38

The smallest positive real variable value is 1.175494E-38. This is the last value that appears in my sequence, which is satisfactory because it decreases like it should do it.

3. In algebra, if p>0 then it is certainly true that 1 + p is different from 1; however, it is not true in a computer. In this machine, if p is positive and small 1 + p may evaluate to

1. The program consists in setting some integer variable to one and then keep dividing it by two until 1 +p is equal 1. The following code estimating the p value where the behavior changes.

```
implicit none
integer :: i
real :: p, sum
p=1.0
do i=1,150
sum=p+1.0
if(sum==1.0 ) then
print*,p
stop
end if
p=p/2.0
end do
stop
end
```

I'll show the output.

5.960464E-08

The p value where the behavior changes is 5.960464E-08. This is the last value in my sequence, which is satisfactory because it decreases like it should do it until the sum of 1 + p is equal 1.

The operating system of my computer is Windows 8.1. The compiler that I am using is Silverfrost FTN95. The exact type of processor in my computer is Intel® Core ™ i3-4030U CPU @ 1.90GHz 1.90GHz.