C programming for beginners

Lesson 2

December 10, 2008

Main task

- What are the values of c that hold

$$x_{n+1} = {x_n}^2 + c \quad (x, c \epsilon \mathbb{C})$$

bounded?

Partial task

- Task 2: For a range of c values, find those for which

$$x_{n+1} = {x_n}^2 + c \quad (x, c \epsilon \mathbb{R})$$

remains bounded.

Program

```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        fclose(of);
        return 0;

}
```

Program

```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        fclose(of);
        return 0;

}
```

Arrays

```
double x[20];
int z[100];
```

Using Arrays

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

Main program

```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        return 0;

}
```

Pointers

```
int *p;
double *p;
char *p;
FILE *p;
```

Main program

```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        return 0;

}
```

Pointers

```
int *p;
double *p;
char *p;
FILE *p;
```

Increasing value

```
(*p)++;
```

Moving the pointer

```
p++;
```

Auxiliary functions

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define ITERS 1000
#define BIG 1000

double recurs (double x, double c){
        return x*x+c;
}

void rdata(char *myfile, double *x){
        FILE *f = fopen(myfile,"r");
        fscanf(f, " %lf %lf %lf", &x[0], &x[1], &x[2]);
        fclose(f);
        return;

}
```

Functions: Argument by value

```
double recurs (double x, double c){
        ....
}
```

Auxiliary functions

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define ITERS 1000
#define BIG 1000

double recurs (double x, double c){
        return x*x+c;
}

void rdata(char *myfile, double *x){
        FILE *f = fopen(myfile,"r");
        fscanf(f, " %lf %lf %lf", &x[0], &x[1], &x[2]);
        fclose(f);
        return;

}
```

Functions: Argument by value

```
double recurs (double x, double c){
        ....
}
```

Functions: Argument por reference

```
void otherfunction(double *x){
        ....
}
```

Auxiliary functions

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define ITERS 1000
#define BIG 1000

double recurs (double x, double c){
        return x*x+c;
}

void rdata(char *myfile, double *x){
        FILE *f = fopen(myfile,"r");
        fscanf(f, " %lf %lf %lf", &x[0], &x[1], &x[2]);
        fclose(f);
        return;

}
```

Functions: Argument by value

```
double recurs (double x, double c){
        ....
}
```

Functions: Argument por reference

```
void otherfunction(double *x){
        ....
}
```

scanf, fscanf

scanf("*format*", &x, &y, ...);
fscanf(*filehandle*, "*format*", &x, &y, ...);

Iterations

```
int iterate(double c, double (*f)(double, double)){
        double x = 0;
        int i = 0;
        while (i<ITERS && x<BIG){
                x = f(x,c);
                i++;
        }
        return i;

}
```

Function pointers

```
int outer_function(double y, double z,
double (*f)(double, double)){
        ...
        x = f(y,z);
        ...
}
```

Iterations

```
int iterate(double c, double (*f)(double, double)){
        double x = 0;
        int i = 0;
        while (i<ITERS && x<BIG){
                x = f(x,c);
                i++;
        }
        return i;

}
```

Function pointers

```
int outer_function(double y, double z,
double (*f)(double, double)){
        ...
        x = f(y,z);
        ...
}
```

Calling the outer function

```
...
if (y < z){
        x = outer_function(y, z, &function1)
} else {
        x = outer_function(y, z, &function2)
}
...
```

Program

```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        fclose(of);
        return 0;

}
```

Data input (from files)

```
double cc[3];
...
char *myfile;
...
myfile = argv[1];
rdata(myfile, cc);
```

Program

```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        fclose(of);
        return 0;

}
```

Data input (from files)

```
double cc[3];
...
char *myfile;
...
myfile = argv[1];
rdata(myfile, cc);
```

File data output

```
FILE *of;
...
of=fopen("result.dat","w");
...
fprintf(of, " %f %d\n", c ,i);
...
fclose(of);
```

Program

```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        fclose(of);
        return 0;

}
```

Iterations

```
for (c=cc[0]; c<cc[1]; c+=cc[2]){
        i=iterate(c,&recurs)
        if (i<ITERS){
        ...
        }
}
```

Program

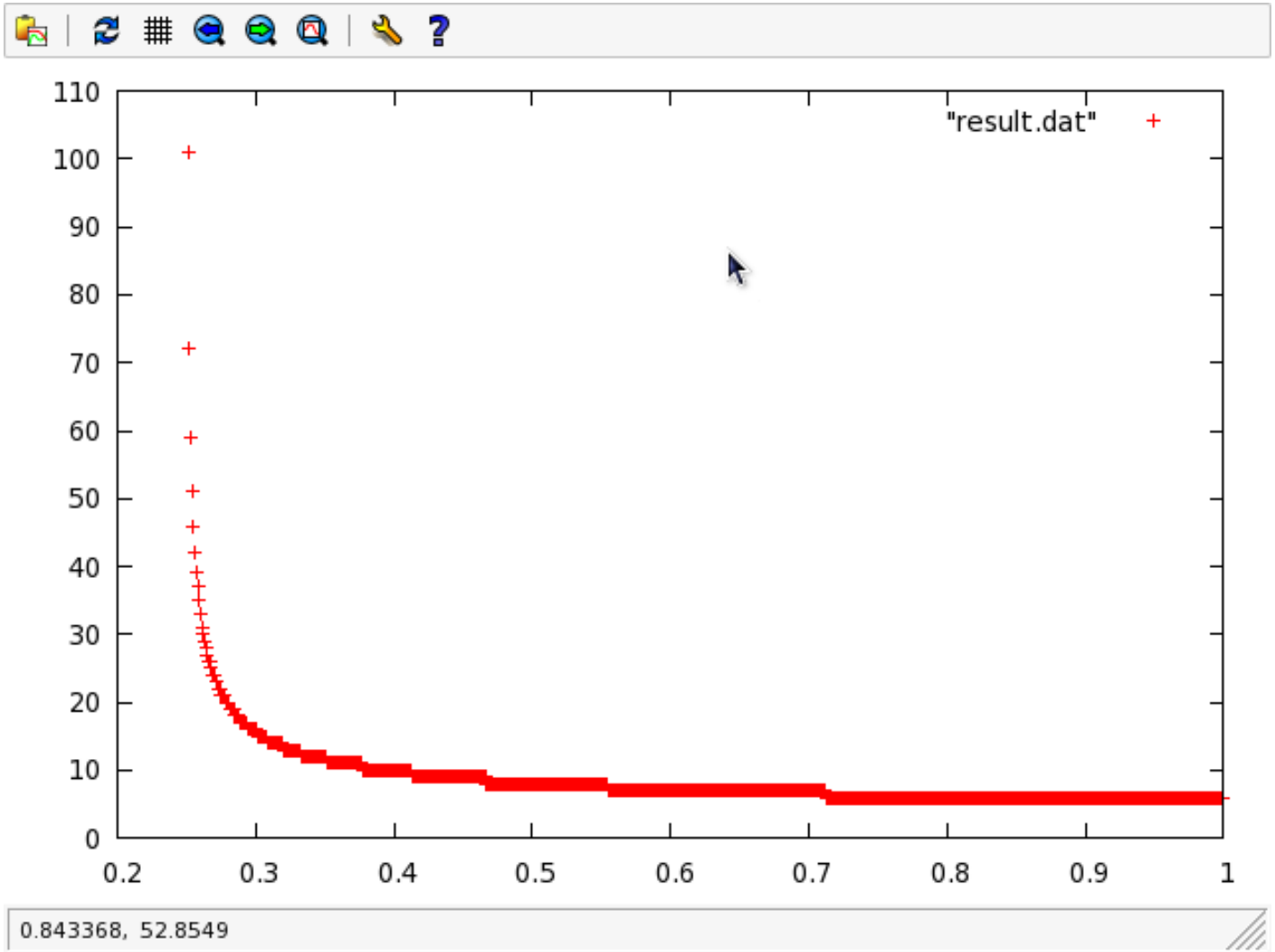
```
int main(int argc, char *argv[]){
        int i;
        double x,c;
        double cc[3];
        char *myfile;
        FILE *of;
        myfile = argv[1];
        rdata(myfile, cc);
        of=fopen("result.dat","w");
        for (c=cc[0]; c<cc[1]; c+=cc[2]){
                if ((i=iterate(c,&recurs))<ITERS){
                        fprintf(of, " %f %d\n", c ,i);
                }
        }
        fclose(of);
        return 0;

}
```

Iterations

```
for (c=cc[0]; c<cc[1]; c+=cc[2]){
        i=iterate(c,&recurs)
        if (i<ITERS){
        ...
        }
}
```

Iterations

```
for (c=cc[0]; c<cc[1]; c+=cc[2]){
        if ((i=iterate(c,&recurs))<ITERS){
        ...
        }
}
```


"result.dat"
110
100
90
80
70
60
50
40
30
20
10
0
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.843368, 52.8549

Summary

What did we learn?

- Arrays
- Pointers
- Function pointers
- Working with data files