C programming for beginners

Lesson 1

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 1: Does

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

remains bounded for a given c value?

Program

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

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

Control structures

while (*condition*){

    ...

}

Program

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

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

Control structures

while (*condition*){

    ...

}

while()

```
while (i<ITERS && x<BIG){
    x = next(x,c);
    i++;
}
```

Program

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

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

Comparison operators

- ==
- < > <= >=
- !=

Program

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

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

Comparison operators

- ==
- < > <= >=
- !=

Logical operators

- &&
- ||
- !

Program

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

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

Comparison operators

- ==
- < > <= >=
- !=

Logical operators

- &&
- ||
- !

Important!

Operators precedence

Program

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

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

Aritmetic operators

- ++ - -
- += -= *= ....

Program

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

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

Aritmetic operators

- ++ - -
- += -= *= ....

Control structures

if (*condition*){

    ...

} else {

    ...

}

Program

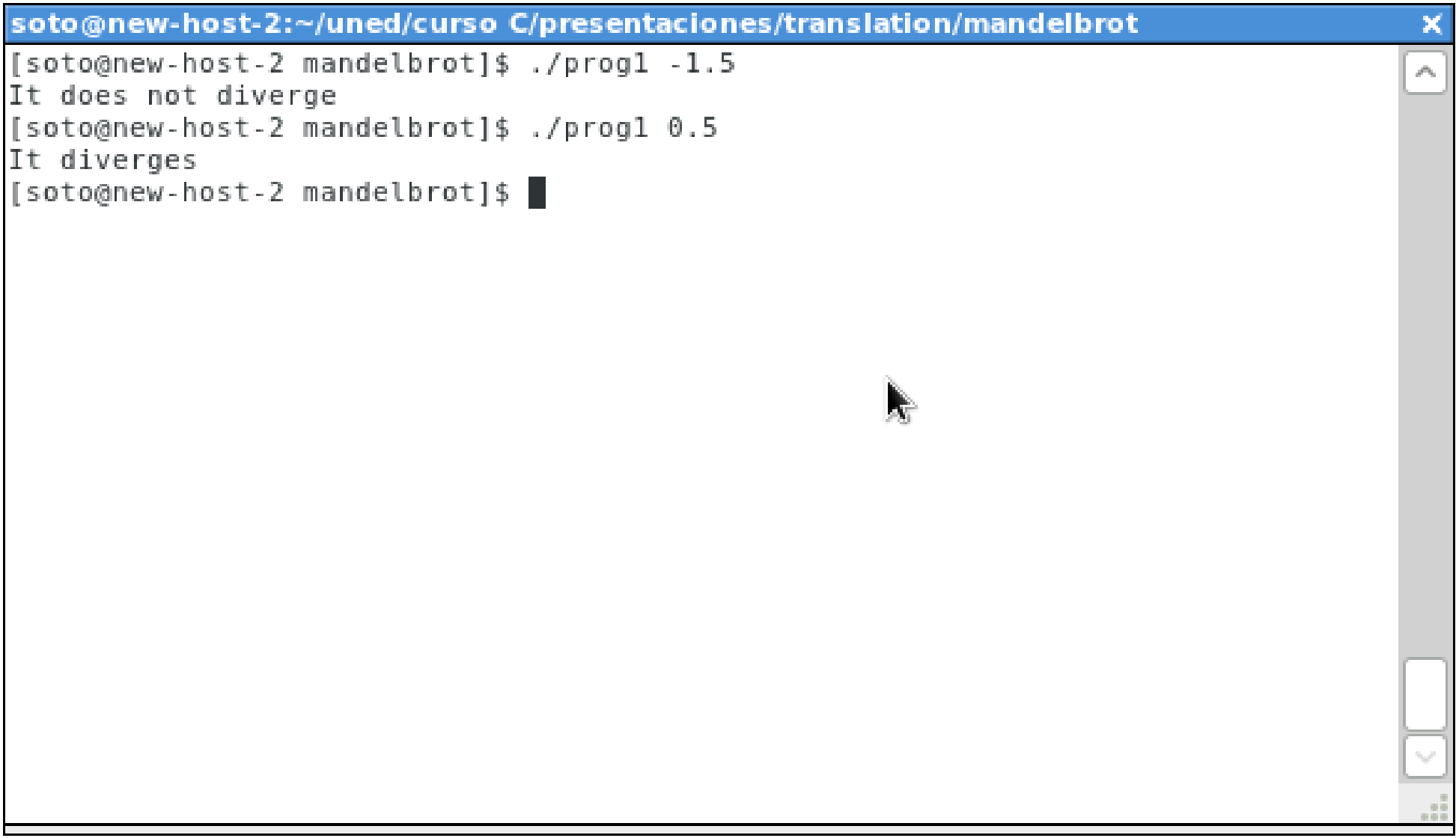
```
#include <stdio.h>
#include <stdlib.h>
#define ITERS 1000
#define BIG 1000

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

int main(int argc, char *argv[]){
        int i,n;
        double x,c;
        c = atof(argv[1]);
        x = 0;
        i = 0;
        while (i<ITERS && x<BIG){
                x = next(x,c);
                i++;
        }
        if (i<ITERS){
                printf("It diverges\n");
        } else {
                printf("It doesn't diverge\n");
        }
        return 0;
}
```

if() ... else

```
if (i<ITERS){
        printf("It diverges\n");
} else {
        printf("It doesn't diverge\n");
}
```


soto@new-host-2:~/uned/curso C/presentaciones/translation/mandelbrot
[soto@new-host-2 mandelbrot]$ ./prog1 -1.5
It does not diverge
[soto@new-host-2 mandelbrot]$ ./prog1 0.5
It diverges
[soto@new-host-2 mandelbrot]$

Summary

What did we learn?

- Comparison operators
- Logical operators
- Increment and decrement operators
- Control structures (while)
- Control structures (if - else)