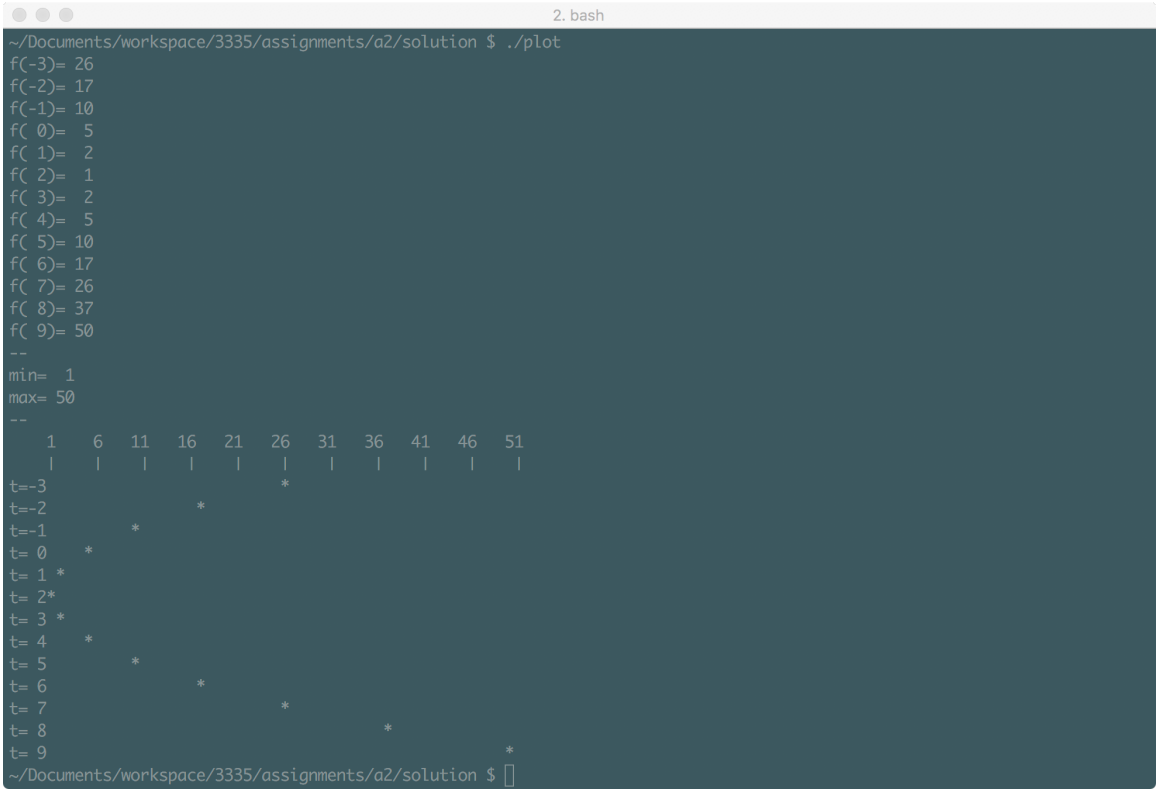
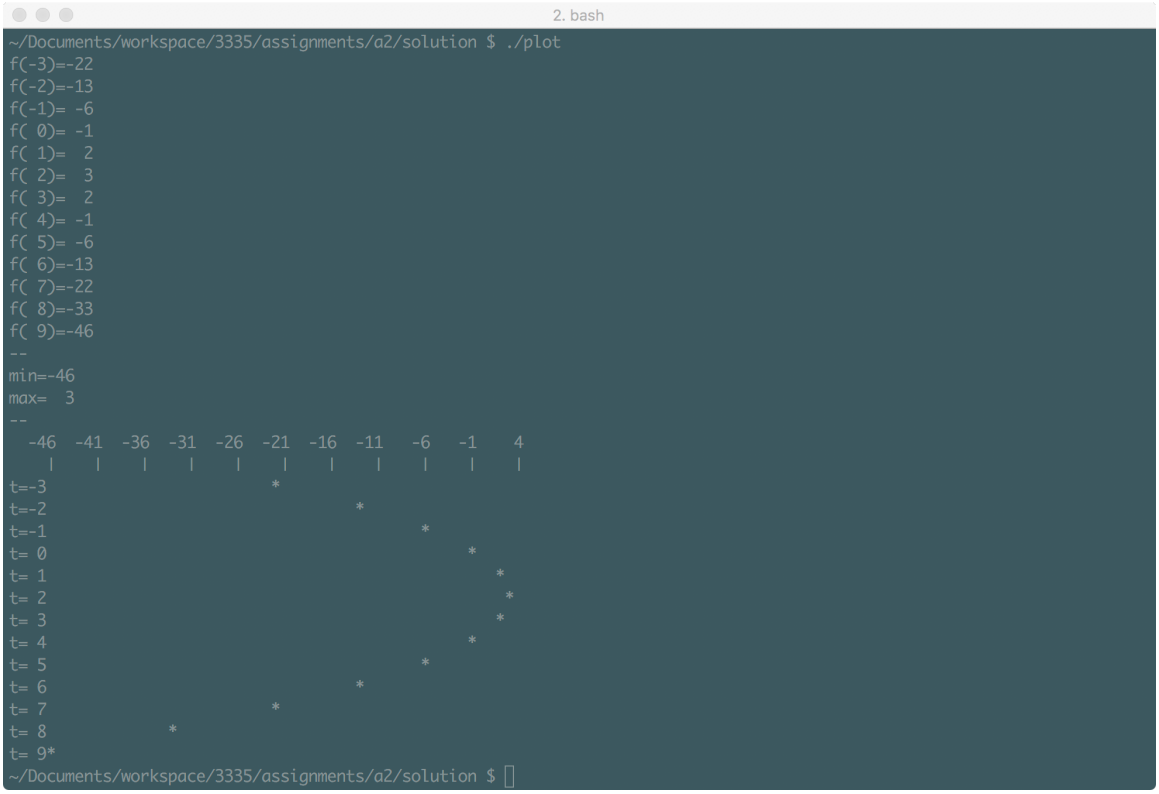
CS3335 Assignment 2 Due: 08/31/17

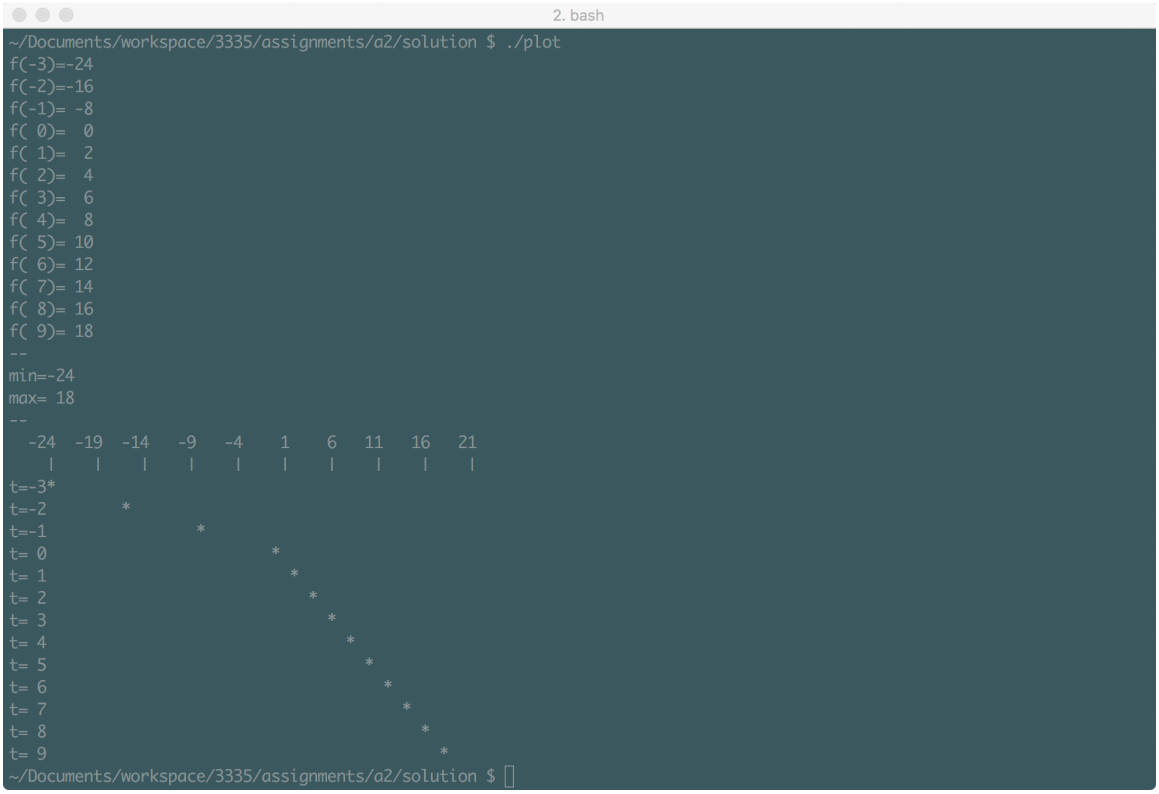
1. (15 points) Complete the attached program `plot.c` to plot a function `f(t)`, e.g. `f(t)=t^2-4t+5` for values of `t` between two values as specified as two variables `low` and `high` on line 11 in `plot.c`. Note, first, nested loops are NOT allowed in your program; second, I am going to change the definition of `f(t)` when testing your program; third, I am going to change the values of `low` and `high` when testing your program.

   Please find a few example `f(t)` shown as comments in the bottom of `plot.c`. The following are their corresponding outputs.




2. bash
~/Documents/workspace/3335/assignments/a2/solution $ ./plot
f(-3)=-22
f(-2)=-13
f(-1)= -6
f(0)= -1
f(1)= 2
f(2)= 3
f(3)= 2
f(4)= -1
f(5)= -6
f(6)=-13
f(7)=-22
f(8)=-33
f(9)=-46
--
min=-46
max= 3
--
-46 -41 -36 -31 -26 -21 -16 -11 -6 -1 4
t=-3
t=-2
t=-1
t= 0
t= 1
t= 2
t= 3
t= 4
t= 5
t= 6
t= 7
t= 8
t= 9*
~/Documents/workspace/3335/assignments/a2/solution $


2. bash
~/Documents/workspace/3335/assignments/a2/solution $ ./plot
f(-3)= -1
f(-2)= -9
f(-1)= -8
f(0)= 0
f(1)= 8
f(2)= 9
f(3)= 1
f(4)= -7
f(5)= -9
f(6)= -2
f(7)= 6
f(8)= 9
f(9)= 4
--
min= -9
max= 9
--
-9 -4 1 6 11
t=-3
t=-2*
t=-1 *
t= 0
t= 1
t= 2
t= 3
t= 4 *
t= 5*
t= 6
t= 7
t= 8
t= 9
~/Documents/workspace/3335/assignments/a2/solution $



Hint:
First, for all `t` between `low` and `high`, find the minimum `f(t)` and maximum `f(t)`, let's call them `f_min` and `f_max` respectively.

Then create an array of characters (i.e. a string) whose size is `f_max-f_min+2`. This is of course taking advantage of the fact that in C, when defining an array, its size is allowed to be a variable. For instance:

```
int m=3*6;
char ex[m];
```

Then for each value of `t`, in the string you created above, store an asterisk in the element whose index is corresponding to the function value `f(t-f_min)`, while all leading elements before the asterisk (if any) are blank and the element at `f(t-f_min+1)` is a `'\0'`. This is of course assuming that the `f(t-f_min)` value is rounded to an integer. Print the string out, clear it up, and go on to the next value of `t`.

2. (10 points) Change the `bitmasks.c` program by **adding the following two functions** and change the `main` function accordingly to test them.

   `unsigned setbits (unsigned x, int p, int n, unsigned y)` that returns x with the n bits that begin at position p (right-adjusted) set to the rightmost n bits of y, leaving the other bits unchanged. Note: **it does not change the values of x and y though**.

   For instance, if x is equal to 2004384122

0111 0111	0111 **1000**	0111 1001	0111 1010

   and y is equal to 1634952294

0110 0001	0111 0011	0110 0100	0110**0110**

   Then `setbits (x, 20, 4, y)` returns

0111 0111	0110 **1100**	0111 1001	0111 1010

   `unsigned invertbits (unsigned x, int p, int n)` that returns x with the n bits that begin at position p (right-adjusted) inverted, i.e. 1 changed to 0 and vice versa, leaving the other bits unchanged. Note: **it does not change the value of x though**.

   For instance, if x is equal to 2004384122

0111 0111	0111 **1000**	**0111 1**001	0111 1010

   Then `invertbits (x, 19, 9)` returns

0111 0111	0111 **0111**	**1000 0**001	0111 1010

What to turn in?
Create a tarball file by the name of `cs3335_a2_yourlastname.tar` that includes

- Completed source code file `plot.c` for question 1.
- Completed source code file `bitmasks.c` for question 2.

Submit the tarball file through BlazeVIEW by the due time.