Brainfuck interpreter

Brainfuck Program Start

* Define “infinite” array all instantiated to 0
  + C equivalent is char array[infinitely large number] = {0};
* Define integer pointer instantiated to beginning of character array
  + C equivalent is char \*ptr = array;

Commands

* > – increment pointer position in array
  + C equivalent is ++ptr;
* < – decrement pointer position in array
  + C equivalent is --ptr;
* + – increment value of the byte currently pointed at
  + C equivalent is ++\*ptr;
* - – decrement value of the byte currently pointed at
  + C equivalent is --\*ptr;
* . – output the byte at the data pointer
  + C equivalent is putchar(\*ptr);
* , – receive one byte of input and store it in data pointer
  + C equivalent is \*ptr = getchar();
* [ – start while loop using current data pointer as counter
  + C equivalent is while (\*ptr) {
* ] – end while loop when counter is 0
  + C equivalent is }

Things to Keep Track of

* Int array to keep all values
* Int to keep track of index value array
* Int to keep track of how much to add / subtract from array index
* Int to keep track of how much to multiply add/subtract if in while loop
* Int array to keep track of number used for while loop
* Int to keep track of index in loop array
* bool to flag if we need to get another character at end of loop
* int we are currently on – cast as character when printing

Handling Operations

* When adding/subtracting continue to go through the next characters and add to int until we come upon a character that isn’t adding
  + If current loop index is larger than 0 multiply add/sub number and every loop number setting
  + Set addsub to 0 after calculations
* When starting a loop keep track of number in loop array and keep track of current index with loop index tracker int
  + If current loop index is greater than 0, move to next index and keep track of number starting loop
* When ending a loop set current loop index to 0
  + If loop index tracker is greater than 0 subtract 1
* When moving through the index in array continue to go through the next characters and add to value index tracker until we come upone a character that isn’t moving through the array
  + If value index tracker is less than 0 report issue and break out of loop
* Don’t do anything if character isn’t one of the recognized operations

Requirement for Program to Run

* Text file of brainfuck program

Psuedo Code / Logic Plans

#include <cstdio>

#include <cstdlib>

#include <fstream>

Int main (argc, argv) {

Check to see if opening file works;

If it doesn’t work, report issue and return;

// declare variables

Int values[100] = {0};

Int valtrack = 0;

Int loop[100] = {0};

Int looptrack = 0;

Int addsub = 0;

Bool flag = true;

Int c = getchar(file);

While (c != EOF) {

Set flag to true

Check if character is not part of the defined operators

// do nothing

Check if we are adding

Set flag to false

// add to addsub

While (((c=getchar()) != EOF) && c==’+’) {

// add to addsub

}

Check if loop[looptrack] > 0 {

addsub \*= loop[looptrack];

For (int I = looptrack-1; I > = 0; i--) {

addsub \*= loop[looptrack]

}

}

Values[valtrack] += addsub

Print out current index, what we added and new value

Check if we are subtracting

Set flag to false

// add to addsub

While (((c=getchar()) != EOF) && c==’-’) {

// add to addsub

}

Check if loop[looptrack] > 0 {

addsub \*= loop[looptrack];

For (int I = looptrack-1; I > = 0; i--) {

addsub \*= loop[looptrack]

}

}

Values[valtrack] -= addsub

Print out current index, what we subtracted and new value

Check if we are moving pointer forward

Set flag to false

While (((c=getchar()) != EOF) && c==’>’) {

// add to valtrack

}

Print out current index

Check if we are moving pointer backward

Set flag to false

While (((c=getchar()) != EOF) && c==’>’) {

// subtract to valtrack

}

Print out current index

Check if we are starting loop

Check if loop[looptrack] >0 {

Add to looptrack

Set loop[looptrack] to current index

}

Else {

Set loop[looptrack] to current index

}

Print out that we started loop [current index] times

Check if we are ending loop

Set loop[looptrack] = 0

Check if looptrack is > 0 {

Subtract looptrack

}

Print out that we ended a loop

Check if we are printing

Print out what we are printing

Check if we are inputting

State that inputs are represented as 999

C=getchar()

}

}