// hw 4

// Ryan Michal

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

*unsigned* *int* reverseBits(*unsigned* *int* x);

*void* shift();

*unsigned* *int* int\_to\_int(*unsigned* *int* k);

*int* main() {

*int* x = reverseBits(7);

printf("Reversed input: %d", x);

shift();

}

*unsigned* *int* reverseBits(*unsigned* *int* x) {

*unsigned* *int* input = x;

*unsigned* *int* reversed = 0;

*unsigned* *int* bits = log2(input)+1 ;

*unsigned* *int* last\_bit;

//go through each bit

for(*int* i = 0; i < bits; i++) {

//shift to last bit & check t

last\_bit = (input & (1 << i));

if(last\_bit) {

//bit was 1; write that into our reversed int

reversed |= (1 << ((bits - 1) - i));

}

}

return reversed;

}

*unsigned* *int* reverseBits(*unsigned* *int* input) {

*unsigned* *int* x = 0;

while(input > 0) {

x<<=1;

if(input & 1 == 1) {

x ^= 1;

}

input >>= 1;

}

return x;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C Compiler.

Code, Compile, Run and Debug C program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

*void* shift();

*unsigned* *int* int\_to\_int(*unsigned* *int* k);

*int* main()

{

printf("Hello World\n");

shift();

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C Compiler.

Code, Compile, Run and Debug C program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

*void* shift()

{

printf("lets smoosh characters together");

*unsigned* *int* product = 0;

*unsigned* *int* a = 'a';

*unsigned* *int* b = 'b';

*unsigned* *int* c = 'c';

*unsigned* *int* d = 'd';

printf("%c %d %d\n", a, a, int\_to\_int(a));

printf("%c %d %d\n", b, b, int\_to\_int(b));

printf("%c %d %d\n", c, c, int\_to\_int(c));

printf("%c %d %d\n", d, d, int\_to\_int(d));

product = product | a;

product = product << 8;

product = product | b;

product = product << 8;

product = product | c;

product = product << 8;

product = product | d;

printf("\n%c%c%c%c = %d", a, b, c, d, product);

}

/\* Credit for int\_to\_int

https://stackoverflow.com/questions/5488377/converting-an-integer-to-binary-in-c?rq=1

\*/

*unsigned* *int* int\_to\_int(*unsigned* *int* k)

{

return (k == 0 || k == 1 ? k : ((k % 2) + 10 \* int\_to\_int(k / 2)));

}

Problem 1:



Problem 2:



PART TWO:

getRate.sh

#!/bin/sh

read -p "Insert rate" rate;

./interest.sh $rate

Interest.sh

#!/bin/sh

# Hw 4 Part 2

# Ryan Michal

# Hello World Program in Bash Shell

#part 2 question 1

# futurevalue=$(bc -l <<< '100 + (1 + 0.05/12) ^ 12');

# echo $futurevalue;

#####################################

#part 2 question 2

# $1 is the first arugment

# this script is called by getrate

futurevalue=$(bc -l <<< "100 + (1 + $1/12) ^ 12");

echo $futurevalue;