TASK 1 Write a program to calculate the overall current in this circuit.

//

// lab 3 task 1

//

// Sidi Liang

//

#include <stdio.h>

#include <math.h>

float Ohm\_law(float voltage, float resistent); //Returns current using the Ohm law

float series(float r1, float r2); //Returns the total resistant of two resistors in a series

float parallel(float r1, float r2); //Returns the total resistant of two resistors in parallel

float parallel\_three(float, float, float); //Returns the total resistant of three resistors in parallel

int main(){

//int r1=100, r2=200, r3=300, r4=400, r5=500, r6=600, r7=700;

int r1=123, r2=234, r3=345, r4=456, r5=567, r6=678, r7=789;

printf("%f\n", Ohm\_law(12, parallel\_three(series(r1, r2), parallel(r3, r4), series(parallel(r6, r7), r5))));

return 0;

}

float Ohm\_law(float voltage, float resistent){

return voltage / resistent;

}

float series(float r1, float r2){

return r1 + r2;

}

float parallel(float r1, float r2){

return 1 / ((1 / r1) + (1 / r2));

}

float parallel\_three(float r1, float r2, float r3){

return 1 / ((1 / r1) + (1 / r2) + (1 / r3));

}

Code for TASK 1

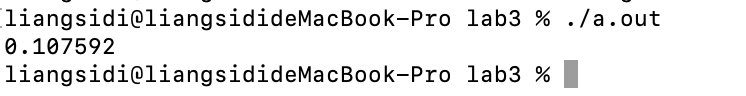


Figure 1: Output of Code in Task 1.

TASK 2 Write a "dice rolling" game. You are probably familiar with 6-sided dice, but some games use dice with 4, 6, 8, 10, 20, and 100 sides!

//

// lab 3 task 2

//

// Sidi Liang

//

#include <stdio.h>

#include <stdlib.h> // extra includes!

#include <time.h>

/\* Get a random number from 0 to 0.9999999

(you don't need to understand this function)

\*\*\*\*\* DON'T MODIFY THIS FUNCTION \*\*\*\*\*

\*/

float getRand() {

return rand() / (RAND\_MAX+1.0);

}

int rollDie(int sides); // pass in number of sides

int main(){

srand( time(NULL) ); // init random

getRand(); // kick-start the random numbers

int value = 0;

value = rollDie(6);

printf("6-sided die: %i\n", value);

value = rollDie(20);

printf("20-sided die: %i\n", value);

getchar();

}

int rollDie(int sides){

float number = getRand();

return (int)((float)sides \* number);

}

Code for TASK 2

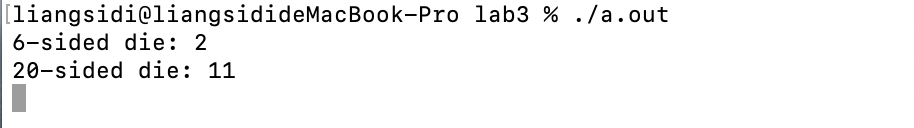


Figure 2: Output of Code in Task 2.

TASK 3 Create a "guessing game" program.

//

// lab 3 task 3

//

// Sidi Liang

//

#include <stdio.h>

#include <stdlib.h> // extra includes!

#include <time.h>

/\* Get a random number from 0 to 0.9999999

(you don't need to understand this function)

\*\*\*\*\* DON'T MODIFY THIS FUNCTION \*\*\*\*\*

\*/

float getRand() {

return rand() / (RAND\_MAX+1.0);

}

int gameSession(int correct\_answer);//our function for the game

int main(){

srand( time(NULL) ); // init random

getRand(); // kick-start the random numbers

int resultFlag = 0;//1 if the user wins, 0 if the user loses

int presetAnswer = (int)(getRand() \* 32 + 0.5);//Generate a ramdom number between 1 and 32

for(int i = 0; i < 5; i++){

if(gameSession(presetAnswer)){//break if returns 1

resultFlag = 1;

break;

}

}

if(resultFlag) puts("you win"); //if result flag is 1, the user wins

else puts("you lose");

return 0;

}

int gameSession(int correct\_answer){

puts("Guess an int between 1 and 32 (inclusive).");//prompt instruction

int userInput;

scanf("%d", &userInput); //input to userInput

if(userInput < correct\_answer){//Compare input with correct\_answer

puts("too low");

return 0;

}

else if(userInput > correct\_answer){

puts("too high");

return 0;

}

else{

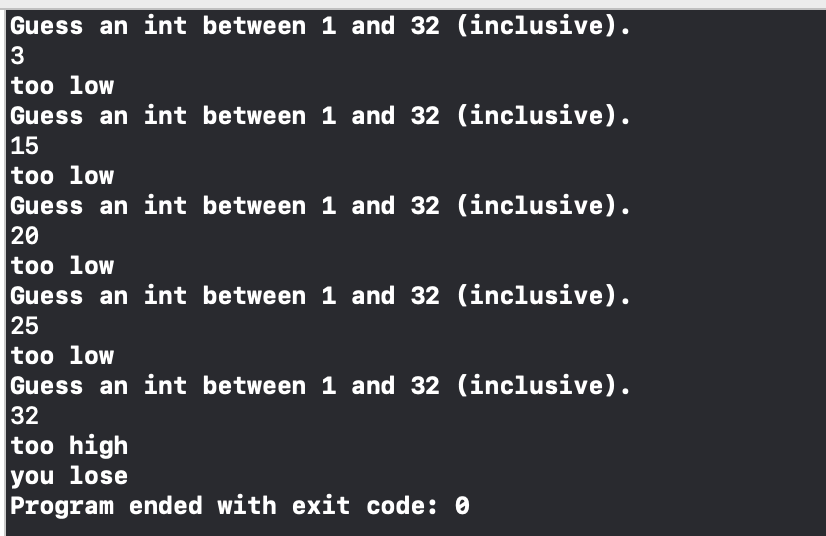
puts("correct");

return 1;

}

}

Code for TASK 3



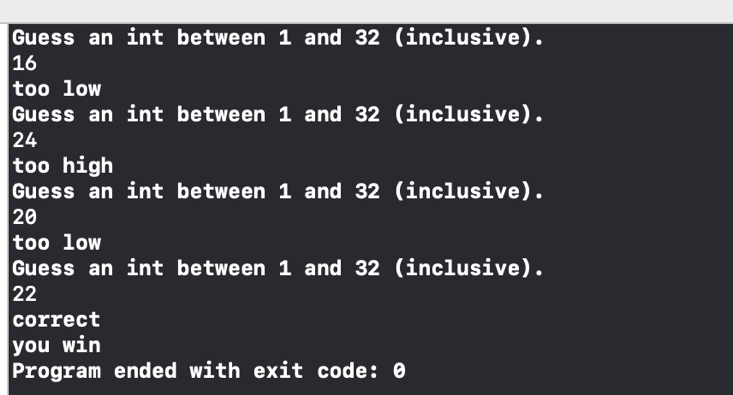


Figure 3: Output of Code in Task 3.