Info Labor 2

Labor 2.1 Bitshifting

#include <stdio.h>  
//#include <math.h>  
int iInteger;  
int iCounter;  
int main(){  
 printf("Enter your Integer\n");  
 scanf("%d",iInteger);  
 iCounter = 0; //Counts the number of occurences where 1 bitshifted by i & the binary number return true  
 int i;  
 for (i=0; i<(sizeof(iInteger))\*8-1; i++) { //i<"...sizeof" refers to the length of the stored binary number  
 //printf("%i\n",i); //Test to see if counter works  
  
 //The following if statement compares 1 bitshifted by i to the binary num stored in iInteger  
 //For example:  
 //iInt = 4 which is stored as 0100 in binary  
 //The for() loop will count i, and the if statement will compare 1 bitshifted by i to the stored integer 0100:  
 //i iInteger (=4) (1<<i) iCounter  
 //0 0100 0001 0  
 //1 0100 0010 0  
 //2 0100 0100 1 -->Because iInteger and (1<<i) both contain a 1 in the same spot, the statement returns  
 //3 0100 1000 1 one, and iCounter is increased by one.  
 //4 --> Program terminates because i=sizeof(iInteger)\*8-1. This formula includes all bits, minus the sign (+ or -)  
 if ((((1 << i)& iInteger)>0) == 1) {  
 iCounter++;  
 }  
 }  
 printf("%i", iCounter);  
 return 0;  
 }

Labor 2.2 Binarzahl Ausgeben

#include <stdio.h>  
  
unsigned int iInteger;  
int bitArray[10];  
  
int main() {  
 printf("Enter your Integer\n");  
 scanf("%d", &iInteger);  
 int i;  
 for (i = 0; i < 31; i++) {  
 if (((iInteger & 1 << i) > 0) == 1) {  
 bitArray[i] = 1;  
 } else {  
 bitArray[i] = 0;  
 }  
 }  
  
 //Finds first zero (?)  
 int firstZero;  
 int b;  
 for (b=0; b<32; b++) {  
 if (bitArray[b]==1){  
 firstZero = b;  
 //printf("firstZero being counted: %d\n",firstZero);  
 }  
 }  
 //Adds one, two, three or no spaces to firstZero to complete first nibblet. Otherwise it could  
 // print incomplete nibblets like 4 = "100"...  
 while ((32-firstZero)%4 != 1){  
 firstZero++;  
 //printf("firstZero becoming even: %d\n", firstZero);  
 }  
 // printf("%d\n", firstZero);  
  
 int breakcounter = 0;  
 for (b = firstZero; b > -1; b--) {  
 breakcounter++;  
 if (breakcounter % 4 == 1) {  
 printf("\n");  
 }  
 printf("%d", bitArray[b]);  
 }  
 return 0;  
}

Labor 2.3 Party Guests

#include <stdio.h>  
int Heinz = 0;  
int Alexander = 1;  
int Frieda = 2;  
int Sophie = 3;  
  
//Use bitshifting to count through the bit, and return every name contained in the bit (1110 = heinz, alex, fried)  
//Enter contained names into the checkcomb. function, which checks if any rules apply  
  
//Function to print binary version of an integer using bitshifting. Creates array with binary number  
int bitArray[4];  
int returnBinaryFifteen(int iInteger) {  
 //int bitArray[4];  
 int i;  
 for (i = 0; i < 4; i++) {  
 if (((iInteger & 1 << i) > 0) == 1) {  
 bitArray[i] = 1;  
 } else {  
 bitArray[i] = 0;  
 }  
 }  
 int b;  
 for (b = 3; b > -1; b--) {  
 printf("%d",bitArray[b]);  
 }  
 printf(" ");  
 return 0;  
}  
int checkCombination() {  
 //Rule one "Mindestens eine Person"  
 int z;  
 int guestCount = 0;  
 for (z=0; z<4; z++) {  
 guestCount += bitArray[z];  
 }  
 if (guestCount == 0){  
 printf(" Violates Rule #1");  
 } else {  
 //Rule two "Heinz und Alexander nicht zusammen"  
 if (bitArray[Heinz] == 1 && bitArray[Alexander] == 1) { //Wenn Heinz (0 place in List) da ist  
 // und Alexander (2nd place) da ist  
 printf(" Violates Rule #2");  
 } else {  
 //Rule Three: "Wenn Frieda geht, dann Sophie auch"  
 if (bitArray[Frieda] == 1 && bitArray[Sophie] == 0) {  
 printf(" Violates Rule #3");  
 } else {  
  
 //Rule Four: "Wenn Heinz und Sophie gehen, dann bleibt Frieda zuhause"  
 if (bitArray[Heinz] == 1 && bitArray[Sophie] == 1 && bitArray[Frieda] == 1) {  
 printf(" Violates Rule #4");  
 } else {  
  
 //Rule Five: "Wenn Heinz zuhause bleibt dann geht entweder Sophie oder Alexander"  
 if (bitArray[Heinz] == 0) { //Wenn Heinz zuhause bleibt  
 if (bitArray[Sophie] == 0 &&  
 bitArray[Alexander] == 0) { //Und Sophie und Alexander zuhause bleiben  
 printf(" Violates Rule #5");  
 }  
 if (bitArray[Sophie] == 1 && bitArray[Alexander] == 1) { //Und Sophie und Alexander beide gehen  
 printf(" Violates Rule #5");  
 }  
 }  
 }  
 }  
 }  
 }  
}  
  
  
int main() {  
 int i;  
 for (i = 0; i < 16; i++) {  
 //Print current number  
 printf("\n");  
 returnBinaryFifteen(i);  
 int b;  
 for (b = 0; b < 4; b++) {  
 if (((1 << b & i) > 0) == 1) {  
 //Heinz  
 if (b == 0) {  
 printf("Heinz ");  
 }  
 //Alexander  
 if (b == 1) {  
 printf("Alexander ");  
 }  
 //Frieda  
 if (b == 2) {  
 printf("Frieda ");  
 }  
 //Sophie  
 if (b == 3) {  
 printf("Sophie ");  
 }  
 }  
 }  
 checkCombination();//Checks rules, and prints violations  
 }  
  
}

Labor 2.4 Quadratic Equation Calc

#include <stdio.h>  
#include <math.h>  
int a;  
int b;  
int c;  
char signZero;  
char signOne;  
char signTwo;  
  
int printScanStoreFunction() {  
 printf("Enter equation following format:\n");  
 printf("+/- ax^2 +/- bx +/- c\n");  
 scanf("%c %dx^2 %c %dx %c %d", &signZero, &a, &signOne, &b, &signTwo, &c);  
 if (((signOne == '+' || signOne== '-') && ((a+b+c) != 0)) && (signTwo == '+' || signTwo == '-')){  
 printf("%c%dx^2 %c %dx %c %d\n",signZero, a, signOne, b, signTwo, c);  
 } else if ((a+b+c) == 0){  
 printf("0x^2+0x+0 is not a valid equation.\n");  
 }else {  
 printf("Input could not be read, please follow structure\n");  
 }  
 //Assign the char signs to the corresponding coefficients:  
 if (signZero == '-') {  
 a = -1\*a;  
 }  
 if (signOne == '-') {  
 b = -1\*b;  
 }  
 if (signTwo == '-') {  
 c = -1\*c;  
 }  
}  
int solveForX () {\  
 //2x = -b +/- sqrt((b\*b - 4\*a\*c)  
 //int insideSqrt = b\*b - 4\*a\*c --> so that if negative different condition applies  
 double insideSqrt = b\*b - 4\*a\*c;  
 /\*printf("a = %d\n",a);  
 printf("b = %d\n",b);  
 printf("c = %d\n",c);\*/  
 double result1;  
 double result2;  
 double resultReal;  
 if (a==0) {  
 printf("Inside a=0");  
 result1 = (-1\*c)/b;  
 printf("Result: %f", result1);  
 }  
 if (insideSqrt > -1 && a>0) {  
 result1 = ((-1\*b) + sqrt(insideSqrt))/(2\*a);  
 result2 = ((-1\*b) - sqrt(insideSqrt))/(2\*a);  
 if (result1==result2) {  
 printf("%f\n",result1);  
 } else{  
 printf("Result:\n%f,%f\n", result1, result2);  
 }  
 } else if (a>0){  
 resultReal = (-1\*b)/(2\*a);  
 result1 = sqrt(-1\*insideSqrt)/(2\*a);  
 result2 = sqrt(-1\*insideSqrt)/(2\*a);  
 printf("Result:\n%g + sqrt(i)\*%g, %g - sqrt(i)\*%g\n", resultReal, result1, resultReal, result2);  
 }  
}  
int main() {  
 printScanStoreFunction();  
 solveForX();  
 return 0;  
}

Labor 2.5

#include <stdio.h>  
  
int main() {  
 int i;  
 int n;  
 for (i=1; i<11; i++){  
 for (n=1; n<11; n++){  
 printf("%d ", (i\*n));  
 if ((i\*n)<10){  
 printf(" ");  
 }  
 }  
 printf("\n");  
 }  
 return 0;  
}

Labor 2.6 Prime Number

#include <stdio.h>  
int number;  
int i;  
int divisibleCount=0;  
int scanPrint(){  
 printf("Enter your Number:\n");  
 scanf("%d", &number);  
}  
int main() {  
 scanPrint();  
 for (i=2; i<11;i++){  
 if (number%i == 0){  
 printf("Number is divisible by %d\n", i);  
 divisibleCount +=1;  
 }  
 }  
 if (divisibleCount == 0){  
 printf("Your number is a prime number");  
 }  
 return 0;  
}

Labor 2.7 Geldautomat

#include <stdio.h>  
  
int accountNumberE;  
int pinE;  
int wrongE;  
  
typedef struct{  
 int accountNumber;  
 int PIN;  
 int moneyLeft;  
 int disabled;  
 int sizeOfName; //one extra for zero  
 char name[];  
}kontoDaten;  
  
int main() {  
 //Konto 1  
 kontoDaten konto1;  
 konto1.accountNumber = 100;  
 konto1.PIN = 5555;  
 konto1.moneyLeft = 1000;  
 konto1.disabled = 0;  
  
 //Konto 2  
 kontoDaten konto2;  
 konto2.accountNumber = 200;  
 konto2.PIN = 5656;  
 konto2.moneyLeft = 1500;  
 konto2.disabled = 0;  
  
 Welcome:  
 printf("Hello, please enter your account number:\n");  
 scanf("%d", &accountNumberE);  
  
 //If Wayne  
 if ((accountNumberE == konto1.accountNumber) && (konto1.disabled == 0)){  
 wrongE = 0;  
 pin1:  
 printf("Hello Wayne, please enter your PIN\n");  
 scanf("%d", &pinE);  
 if (pinE==konto1.PIN){  
 printf("Account Access Granted\n");  
 printf("Money Left:%d\n",konto1.moneyLeft);  
 } else{  
 printf("PIN is wrong!\n");  
 wrongE ++;  
 if (wrongE>2){  
 printf("Account has been disabled due to repeated false attempts\n");  
 konto1.disabled=1;  
 goto Welcome;  
 }  
 goto pin1;  
 }  
 } else if (konto1.disabled ==1){  
 printf("Account has been disabled due to repeated false attempts\n");  
 goto Welcome;  
 }  
 //If Garth  
 if ((accountNumberE == konto2.accountNumber) && (konto2.disabled == 0)){  
 wrongE = 0;  
 pin2:  
 printf("Hello Garth, please enter your PIN\n");  
 scanf("%d", &pinE);  
 if (pinE==konto2.PIN){  
 printf("Account Access Granted\n");  
 printf("Money Left:%d\n",konto2.moneyLeft);  
 } else{  
 printf("PIN is wrong!\n");  
 wrongE ++;  
 if (wrongE>2){  
 printf("Account has been disabled due to repeated false attempts\n");  
 konto2.disabled=1;  
 goto Welcome;  
 }  
 goto pin2;  
 }  
 } else if (konto2.disabled ==1){  
 printf("Account has been disabled due to repeated false attempts\n");  
 goto Welcome;  
 }  
 //If not recognised  
 if ((accountNumberE != konto2.accountNumber) && (accountNumberE != konto1.accountNumber)) {  
 printf("Account not found. Please try again.\n");  
 goto Welcome;  
 }  
 return 0;  
}