Program 1:

//Toni Hunter 180009925

//Assignment 2: Project 1

//

//Write a program that will produce 2 randomized numbers to multiply, then asks user for answer input

//if input is correct, another question is asked and if incorrect, user is asked repeatedly for the answer

//until correct

//

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int mult(void); //function prototype

int main(void)

{

srand(time(NULL));

int user; //user's answer

int answer = mult(); // multiplies numbers

scanf("%d", &user); //first scan

while (user != -1) //sentinel

{

while (user != answer){

printf("No. Please try again!\n");

scanf("%d", &user);

}

if (user == answer){ //if user is correct

printf("Very good!!");

answer = mult();

scanf("%d", &user);

}

}

if (user == -1){ // sentinel reached

printf("Keep up the good work!");

}

}

int mult(void)

{

int user; //user's answer

int num1 = (10 + rand() %89); //double digits only

int num2 = (10 + rand() %89);

printf("How much is %d times %d?\n" ,num1, num2); //first question

return num1 \* num2; // return product of two numbers

}

Program 2:

// Toni Hunter 180009925

//Assignment 2: Project 2

//

//Simulating the game of craps. except adding a wager for user

//

#include <stdio.h>

#include <stdlib.h>

#include <time.h> // contains prototype for function time

// enumeration constants represent game status

enum Status { CONTINUE, WON, LOST };

int rollDice(void); // function prototype

//int Craps(void); //function prototype

int main(void)

{

int Wager;

int Balance = 2000; //initial bank balance

printf("Rules to Craps!:\n1.Two dices are required to play and a player rolls two six-sided dice and adds the numbers rolled together.\n2.If on the first roll a player encounters a total of 7 or 11 the player automatically wins,\nand if the player rolls a total of 2, 3, or 12 the player automatically loses, and play is over.\n3.If a player rolls a total of 4, 5, 6, 8, 9, or 10 on their first roll, that number becomes the point.\nThen the player continues to roll the two dice again until one of two things happens \neither they roll the point again, in which case they win, or they roll a 7, in which case they lose.\n");

printf("\nYour initial bank balance is $%d\n", Balance);

printf("please Enter your wager: ");

scanf("%d", &Wager);

printf("\n");

while (Wager > Balance)

{

printf("\nWager is too high, please enter a lower wager: ");

scanf("%d", &Wager);

}

while (Wager != -1) // sentinel

{

int Crap = Craps();

if (Crap == 1) //win

{

printf("\nYour Balance is now $%d. Now we're talking!\n", Balance += Wager);

}

if (Crap == 2) //lose

{

printf("Your Balance is now $%d. Want to keep going?\n", Balance -= Wager);

}

if (Balance == 0){ //breaks out of game is player balance = 0

printf("Sorry, you're broke!\n");

break;

}

printf("\nPlease enter new wager to play again or -1 to quit: "); // choice for player

scanf("%d", &Wager);

while (Wager > Balance)

{

printf("\nWager is too high, please enter a lower wager: ");

scanf("%d", &Wager);

}

}

printf("\nYour total winnings are $%d. Play again sometime!", Balance);

}

int Craps(void)

{

// randomize random number generator using current time

srand(time(NULL));

int myPoint; // player must make this point to win

enum Status gameStatus; // can contain CONTINUE, WON, or LOST

int sum = rollDice(); // first roll of the dice

// determine game status based on sum of dice

switch(sum) {

// win on first roll

case 7: // 7 is a winner

case 11: // 11 is a winner

gameStatus = WON;

break;

// lose on first roll

case 2: // 2 is a loser

case 3: // 3 is a loser

case 12: // 12 is a loser

gameStatus = LOST;

break;

// remember point

default:

gameStatus = CONTINUE; // player should keep rolling

myPoint = sum; // remember the point

printf("Point is %d\n", myPoint);

break; // optional

}

// while game not complete

while (CONTINUE == gameStatus) { // player should keep rolling

sum = rollDice(); // roll dice again

// determine game status

if (sum == myPoint) { // win by making point

gameStatus = WON;

}

else {

if (7 == sum) { // lose by rolling 7

gameStatus = LOST;

}

}

}

// display won or lost message

if (WON == gameStatus) { // did player win?

puts("Player wins");

return gameStatus;

}

else { // player lost

puts("Player loses");

return gameStatus;

}

}

// roll dice, calculate sum and display results

int rollDice(void)

{

int die1 = 1 + (rand() % 6); // pick random die1 value

int die2 = 1 + (rand() % 6); // pick random die2 value

// display results of this roll

printf("Player rolled %d + %d = %d\n", die1, die2, die1 + die2);

return die1 + die2; // return sum of dice

}

Program 3:

//Toni Hunter 180009925

//Assignment 2: Program 3

//

//Write a recursive function that takes a number(m) and adds to itself n amount of times.

//assume the integer is greater than or equal to one.

//

#include <stdio.h>

#include <stdlib.h>

int Multiply(num1, num2)

{

//base case

if (num2 == 0){

return num2;

}

//recursive call

else{

return num1 + Multiply(num1, num2 - 1);

}

}

int main(void)

{

int num1;

int num2;

printf("Enter first positive integer (multiplicand): "); //ex.4 3 + 3 ...

scanf("%d", &num1);

printf("Enter second positive integer (multiplier): "); // ex. 4

scanf("%d", &num2);

printf("%d \* %d = %d", num1, num2, Multiply(num1, num2));

return 0;

}

Program 4:

//Toni Hunter 180009925

//Assignment 2: Program 4

//

//A program that simulates coin tossing 50,000 times. Heads and tails is recorded and printed out

//and also a percentage of each is printed out.

//

#include <stdio.h>

#include <stdlib.h>

#include <time.h> //function prototype for time

int main()

{

srand(time(NULL)); // randomizes based on clock to obtain seed

int frequencyH = 0; //initialized tossed head counter

int frequencyT = 0; //initialized tosses tails counter

float percentageH;

float percentageT;

for (unsigned int toss = 1; toss <= 50000; toss++){

int coin = 1 + rand() % 2; //random number 1 to 2 since H and T

switch(coin){

case 1: //tossed head

frequencyH++;

break;

case 2: //tossed tails

frequencyT++;

break;

}

}

percentageH = (frequencyH/50000.0) \* 100; //.0 makes float

percentageT = (frequencyT/50000.0) \* 100;

//formatting FOR USER FRIENDLINESS!!

printf(" %s|%11s|%11s\n", "Side", "frequency", "Percentage");

printf("-----|-----------|------------\n");

printf("Tails| %10u|%10.2f\n", frequencyT, percentageT);

printf("Heads| %10u|%10.2f\n", frequencyH, percentageH);

}