DESIGN

Include libraries : <stdio.h> and <stdlib.h>

Defining global variables :

const char * names[]	# The names of all the players
typedef enum faciem	# Enumerate all possbile die outcomes
faces die[]	# Make an array of those outcomes
money_in_pot = 0;	# We make a variable to track money in the pot

* Definition of local variables will be shown in the pseudocode.

Defining functions:-

This program will have 5 functions in total:

1) main :- This function takes seed and numbers of players as input from user and calls start game function to begin the game.

2) start_game:- This function contains the main code and calls other functions as required.

3) output_handler:- This function takes the outcome of the die after the die has been rolled and then switches to the relevant case according to the die outcome it received and does certain functions.

4) left :- This function has been borrowed from the asgn.pdf provided in class and it calculates the position of the player to the left of the current player

5) right :- his function has been borrowed from the asgn.pdf provided in class and it calculates the position of the player to the right of the current player.

PSEUDOCODE:-

#include <stdio.h>
#include <stdlib.h>

Define global varibles as mentioned above

Define left function (current player , total players playing)
{

Calculates and returns the position of player to the left
}

Define right function (current player , total players playing)
{

Calculates and returns the position of player to the right
}

Define output_handler function (dice roll outcome, current player , total players playing, money_array[])
{

Initialize Left_or_right_player_variable

Switch (die roll outcome)
{

case Left(0):
add money to player on left to current player
deduct money from current player
print the action that was performed
break

case Right(1):
add money to player on right to current player
deduct money from current player
print the action that was performed
break

case Center(0):
add money to the pot i.e increament money_in_pot variable
deduct money from current player
print the action that was performed
break

case Pass(3,4 or 5):
print the action that was performed i.e player gets a pass
break
}

Return from function
}

Define start_game function (seed , total number of players playing)

{

srand(seed) # Provide the seed to srand before calling rand() for a particular outcome

Initialize outcome variable. # To store outcome of die roll during each player's turn

Initialize check_counter variable # Counter used while checking the bank of players

Initialize counter variable # To keep track of the which player is currently rolling

If (number of players if atleast 2 and maximum 10)
{

Initialize money_array[size equal to the total number of players]

for loop (LOOP FOR ALL PLAYERS)
{
Fill money_array with 3 as each player starts with 3 dollars
i.e [3,3,3…..]
}

while loop (Infinite)
{

initiliaze coun variable # To count number of zeroes during bank check of players

for loop (start with player 0, if player < total players, next player)
{
if ()
Check if money_array has only 1 player left with money in bank
if yes then exit else continue (Use 'coun' to count zeroes in money_array)

else
for loop (start with player 0, if player < total players, next player)
{
Initialiaze roll_times varibles to keep track of rolls remaning

if (player has 3 or more dollars):-

while loops thrice for players with 3 dollars
{
coun = 0
outcome **=** roll dice
for each roll first check if more than 1 player has more than 0 dollars
if yes only then call output_handler
increament roll_times
else print winner and break
}

elif (player has 2 dollars):-
while loops twice for players with 2 dollars
{
coun = 0
outcome **=** roll dice
for each roll first check if more than 1 player has more than 0 dollars
if yes only then call output_handler
increament roll_times
else print winner and break
}
elif (player has 1 dollar)
{
coun = 0
outcome **=** roll dice
for each roll first check if more than 1 player has more than 0 dollars
if yes only then call output_handler
increament roll_times
else print winner and break

}
}

}

}

}

Return from function
}

Define main function ()
{

Initialize seed and number of players variable

print “Random seed: ”
ask for input and store in seed variable

print "How many players? “
ask for input and store in number of players variable

call start_game function (seed , number of players)

Return from function

}
