The “Triple Roll” game

The ACME Computing Corporation is a small company that produces fun software apps. You have been hired by ACME Corp to produce a small gaming application. It has been decided that the software will be written in C. You have been asked to produce a working prototype for ACME’s new “Triple Roll” dice game.

The rules for “Triple Roll” are as follows:

- The user has 3 ordinary dice, each with 6 sides with numbers 1 through to 6 (inclusive) on them. The user throws the 3 dice at once. This can be simulated using a random number generator.
- The user bets on the outcome of the 3 dice in a series of rounds, using a virtual currency called “credits”. The value of credits that the user has should be shown on the screen as each round progresses.
- At the beginning of the game, the user has 100 credits.
- At each round, the user can choose to either bet a certain amount of credits or quit the game. The user cannot bet more than the amount of credit they have. If the user ends up with 0 credits at the beginning of a round, and is thus unable to place a bet, then the game ends.
- For each round, the values of the 3 dice will determine whether the user has won credits, or lost as follows:

    - **ACES:** If all the 3 dice show a 1, the user wins 30 times their bet plus their original stake back
    - **TRIPLE:** If the 3 dice all show three same values (other than a 1 as above) then the user wins 20 times their bet plus their original stake back.
    - **PAIR:** If 2 dice show the same value and the third is different, then the user wins 6 times their bet plus their original stake back.
    - **HIGHS:** If none of these prizes have been won, if the total of the dice come to 15 or more, then the user wins 2 times their bet plus their original stake back.
    - **ODDS:** If one dice shows a 6, or all the values of the dice are odd numbers, and the user has not yet got their bet back by winning another prize then the user gets their bet back.
    - **LOSS:** In all other situations, the user loses their bet and the value of the bet is deducted from their current credit.

Implementation (70%):

- Main program loop and user interaction - prompting the user for input, collecting inputs, and informing the user of the outcomes of throwing the dice and the result of the round. (20%)
- Implementation of pay out cases. (30% - 5% each)
- Quality of coding – variables, comments, functions. (20%)

Test (10%):

- Evidence of a working program with user friendly interaction - prompts, messages, updates on the amount of credits. (10%)