

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

//Constantes
#define MAIOR_VALOR_DE_UM_DADO 6
#define MENOR_VALOR_DE_UM_DADO 1

#define MENOR_VALOR_DE_APOSTA_POSSIVEL 1.0
#define VALOR_MAXIMO_DE_FINALIZACAO_DO_JOGO 0

#define NUMERO_DA_PRIMEIRA_PARTIDA 1

#define TAMANHO_DO_VETOR_DE_VALORES_DE_DADO_QUE_SAIRAM 6

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis locais
    double valorApostado = 0.0;
    double valorGanhoNaPartida = 0.0;
    double saldoDoJogador = 0.0;

    int valorDoPrimeiroDadoDoJogador;
    int valorDoSegundoDadoDoJogador;

    int valorDoPrimeiroDadoDoCroupier;
    int valorDoSegundoDadoDoCroupier;

    bool dadosDeEntradaValidos = true;
    bool sairDaColetaDeDadosDoJogo = false;

    int quantidadeDePartidas = 0;
    int quantidadeDePartidasSemGanhos = 0;

    double maiorValorGanhoEmUmaPartida = MENOR_VALOR_DE_APOSTA_POSSIVEL;

    int contadoresDeValoresDeDadoQueSairam[TAMANHO_DO_VETOR_DE_VALORES_DE_DADO_QUE_SAIRAM];

    int maiorValorDeDadoQueSaiu = 0;

    //zerar vetor de contadores de valores dos dados que sairam
    for (int i = 0; i < TAMANHO_DO_VETOR_DE_VALORES_DE_DADO_QUE_SAIRAM; i++)
    {
        contadoresDeValoresDeDadoQueSairam[i] = 0;
    }

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa Jogo de Dados>>");
    printf("\n\n ## Autor: Leonardo Melo");

    do{
        do{
            dadosDeEntradaValidos = true;

            printf_s("\n\n\nDigite o valor apostado (R$): ");
            scanf_s("%lf", &valorApostado);

            fflush(stdin);

            if (valorApostado <= VALOR_MAXIMO_DE_FINALIZACAO_DO_JOGO)
            {
                sairDaColetaDeDadosDoJogo = true;
                break;
            }

            if (valorApostado < MENOR_VALOR_DE_APOSTA_POSSIVEL)
            {
```

```
        printf_s("\n\nEntrada Invalida!\nO valor apostado nao pode ser menor que R$ %.2lf.",  
MENOR_VALOR_DE_APOSTA_POSSIVEL);  
  
        dadosDeEntradaValidos = false;  
    }  
  
    } while (!dadosDeEntradaValidos);  
  
    if (sairDaColetaDeDadosDoJogo)  
    {  
        break;  
    }  
  
    quantidadeDePartidas++;  
  
    saldoDoJogador -= valorApostado;  
  
    do{  
        dadosDeEntradaValidos = true;  
  
        printf_s("\n\nDigite o valor do primeiro dado do jogador: ");  
        scanf_s("%d", &valorDoPrimeiroDadoDoJogador);  
  
        fflush(stdin);  
  
        if (valorDoPrimeiroDadoDoJogador > MAIOR_VALOR_DE_UM_DADO || valorDoPrimeiroDadoDoJogador <  
MENOR_VALOR_DE_UM_DADO)  
        {  
            printf_s("\n\nEntrada Invalida!\nO valor do dado nao pode ser menor que %d ou maior que %d.",  
MENOR_VALOR_DE_UM_DADO, MAIOR_VALOR_DE_UM_DADO);  
  
            dadosDeEntradaValidos = false;  
        }  
    } while (!dadosDeEntradaValidos);  
  
    contadoresDeValoresDeDadoQueSairam[valorDoPrimeiroDadoDoJogador - 1]++;  
  
    do{  
        dadosDeEntradaValidos = true;  
  
        printf_s("\n\nDigite o valor do segundo dado do jogador: ");  
        scanf_s("%d", &valorDoSegundoDadoDoJogador);  
  
        fflush(stdin);  
  
        if (valorDoSegundoDadoDoJogador > MAIOR_VALOR_DE_UM_DADO || valorDoSegundoDadoDoJogador <  
MENOR_VALOR_DE_UM_DADO)  
        {  
            printf_s("\n\nEntrada Invalida!\nO valor do dado nao pode ser menor que %d ou maior que %d.",  
MENOR_VALOR_DE_UM_DADO, MAIOR_VALOR_DE_UM_DADO);  
  
            dadosDeEntradaValidos = false;  
        }  
    } while (!dadosDeEntradaValidos);  
  
    contadoresDeValoresDeDadoQueSairam[valorDoSegundoDadoDoJogador - 1]++;  
  
    do{  
        dadosDeEntradaValidos = true;  
  
        printf_s("\n\nDigite o valor do primeiro dado do croupier: ");  
        scanf_s("%d", &valorDoPrimeiroDadoDoCroupier);  
  
        fflush(stdin);  
  
        if (valorDoPrimeiroDadoDoCroupier > MAIOR_VALOR_DE_UM_DADO || valorDoPrimeiroDadoDoCroupier <  
MENOR_VALOR_DE_UM_DADO)
```

```

        {
            printf_s("\n\nEntrada Invalida!\n0 valor do dado nao pode ser menor que %d ou maior que %d.",
MENOR_VALOR_DE_UM_DADO, MAIOR_VALOR_DE_UM_DADO);

            dadosDeEntradaValidos = false;
        }

    } while (!dadosDeEntradaValidos);

    contadoresDeValoresDeDadoQueSairam[valorDoPrimeiroDadoDoCroupier - 1]++;

    do{
        dadosDeEntradaValidos = true;

        printf_s("\n\nDigite o valor do segundo dado do croupier: ");
        scanf_s("%d", &valorDoSegundoDadoDoCroupier);

        fflush(stdin);

        if (valorDoSegundoDadoDoCroupier > MAIOR_VALOR_DE_UM_DADO || valorDoSegundoDadoDoCroupier <
MENOR_VALOR_DE_UM_DADO)
        {
            printf_s("\n\nEntrada Invalida!\n0 valor do dado nao pode ser menor que %d ou maior que %d.",
MENOR_VALOR_DE_UM_DADO, MAIOR_VALOR_DE_UM_DADO);

            dadosDeEntradaValidos = false;
        }

    } while (!dadosDeEntradaValidos);

    contadoresDeValoresDeDadoQueSairam[valorDoSegundoDadoDoCroupier - 1]++;

    bool ganhouAlgumValor = false;

    //resultado da partida (Calculos)
    if ((valorDoPrimeiroDadoDoJogador == valorDoPrimeiroDadoDoCroupier && valorDoSegundoDadoDoJogador ==
valorDoSegundoDadoDoCroupier)
        || (valorDoPrimeiroDadoDoJogador == valorDoSegundoDadoDoCroupier && valorDoSegundoDadoDoJogador ==
valorDoPrimeiroDadoDoCroupier))
    {
        valorGanhoNaPartida = valorApostado * 6;
        ganhouAlgumValor = true;

        saldoDoJogador += valorGanhoNaPartida;

        if (maiorValorGanhoEmUmaPartida < valorGanhoNaPartida)
        {
            maiorValorGanhoEmUmaPartida = valorGanhoNaPartida;
        }

        printf_s("\n\n0 jogador ganhou R$ %.2lf.", valorGanhoNaPartida);
    }

    if (!ganhouAlgumValor)
    {
        if ((valorDoPrimeiroDadoDoJogador + valorDoSegundoDadoDoJogador) == (valorDoPrimeiroDadoDoCroupier +
valorDoSegundoDadoDoCroupier))
        {
            valorGanhoNaPartida = valorApostado * 3;
            ganhouAlgumValor = true;

            saldoDoJogador += valorGanhoNaPartida;

            if (maiorValorGanhoEmUmaPartida < valorGanhoNaPartida)
            {
                maiorValorGanhoEmUmaPartida = valorGanhoNaPartida;
            }

            printf_s("\n\n0 jogador ganhou R$ %.2lf.", valorGanhoNaPartida);

```

```
    }
}

if (!ganhouAlgumValor)
{
    if (((valorDoPrimeiroDadoDoJogador + valorDoSegundoDadoDoJogador) % 2 == 0) && (
(valorDoPrimeiroDadoDoCroupier + valorDoSegundoDadoDoCroupier) % 2 == 0)
        || ((valorDoPrimeiroDadoDoJogador + valorDoSegundoDadoDoJogador) % 2 != 0) && (
(valorDoPrimeiroDadoDoCroupier + valorDoSegundoDadoDoCroupier) % 2 != 0))
    {
        valorGanhoNaPartida = valorApostado / 2;
        ganhouAlgumValor = true;

        saldoDoJogador += valorGanhoNaPartida;

        if (maiorValorGanhoEmUmaPartida < valorGanhoNaPartida)
        {
            maiorValorGanhoEmUmaPartida = valorGanhoNaPartida;
        }

        printf_s("\n\n0 jogador ganhou R$ %.2lf.", valorGanhoNaPartida);
    }
}

if (!ganhouAlgumValor)
{
    valorGanhoNaPartida = 0.0;

    quantidadeDePartidasSemGanhos++;

    printf_s("\n\n0 jogador perdeu.");
}

} while (true);

printf_s("\n\n\nJogo Encerrado!");

printf_s("\n\n\n0 jogador jogou %d vez(es).", quantidadeDePartidas);
printf_s("\n\n0 saldo final foi de %.2lf reais.", saldoDoJogador);

for (int i = 0; i < TAMANHO_DO_VETOR_DE_VALORES_DE_DADO_QUE_SAIRAM; i++)
{
    if (contadoresDeValoresDeDadoQueSairam[i] > maiorValorDeDadoQueSaiiu)
    {
        maiorValorDeDadoQueSaiiu = contadoresDeValoresDeDadoQueSairam[i];
    }
}

for (int i = 0; i < TAMANHO_DO_VETOR_DE_VALORES_DE_DADO_QUE_SAIRAM; i++)
{
    if (contadoresDeValoresDeDadoQueSairam[i] == maiorValorDeDadoQueSaiiu)
    {
        printf_s("\n\n0 valor do dado que mais saiu foi %d, %d vez(es).", i + 1,
contadoresDeValoresDeDadoQueSairam[i]);
    }
}

printf_s("\n\n0 maior valor ganho em uma partida foi %.2lf reais.", maiorValorGanhoEmUmaPartida);
printf_s("\n\n0 jogador perdeu %d vez(es).", quantidadeDePartidasSemGanhos);

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```