using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Buoi3

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

int LuaChon = -1;

do

{

Console.WriteLine("1.Tính Giai Thừa:");

Console.WriteLine("2. Tính Lũy Thừa: ");

Console.WriteLine("3. USCLN: ");

Console.WriteLine("4. Chuyển đổi nhị phân: ");

Console.WriteLine("5. Tính số đôi thỏ: ");

Console.WriteLine("6. Tính S(n) = 1 + 3 + 5....+(2n-1)");

Console.WriteLine("7. Tinh S(n) = 1 - 2 = 3 - 4 ....(-1)^(n + 1) \* n");

Console.WriteLine("8. 1 + 1/2 + 1/ 2\*4 + 1 / 2\*4\*6 +...+ 1 / 2\*4\*6\*...2n : ");

Console.WriteLine("0. Thoát ");

Console.Write("Nhập lựa chọn: ");

LuaChon = int.Parse(Console.ReadLine());

switch (LuaChon)

{

case 1:

{

Console.Write("Nhập n: ");

ulong n = ulong.Parse(Console.ReadLine());

ulong kq = tinhGiaiThua(n);

Console.WriteLine($"{n}! kết quả là: {kq} ");

Console.WriteLine();

break;

}

case 2:

{

Console.Write("Nhập a: ");

int a = int.Parse(Console.ReadLine());

Console.Write("Nhập n: ");

int n = int.Parse(Console.ReadLine());

int kq = LuyThua(a, n);

Console.WriteLine($"{a} \* {n} = {kq}");

Console.WriteLine();

break;

}

case 3:

{

Console.WriteLine("Nhập a: ");

int a = int.Parse(Console.ReadLine());

Console.WriteLine("Nhập b: ");

int b = int.Parse(Console.ReadLine());

int kq = USCLN(a, b);

Console.WriteLine($"{a} , {b} = {kq}");

Console.WriteLine();

break;

}

case 4:

{

Console.WriteLine("Nhập n: ");

int n = int.Parse(Console.ReadLine());

string kq = doiNhiPhan(n);

Console.WriteLine($"Nhị phân của {n} là {kq} ");

Console.WriteLine();

break;

}

case 5:

{

Console.Write("Nhập số đôi thỏ: ");

int n = int.Parse(Console.ReadLine());

int kq = soDoiTho(n);

Console.Write($"Số đôi thỏ sau {n} tháng là : {kq} đôi");

Console.WriteLine();

break;

}

case 6:

{

Console.WriteLine("Nhập số nguyên dương n: ");

int n = int.Parse(Console.ReadLine());

int kq = tongS(n);

Console.Write($"Tổng S{n} = {kq}");

Console.WriteLine();

break;

}

case 7:

{

Console.WriteLine(" Nhập số nguyên dương n: ");

int n = int.Parse(Console.ReadLine());

int kq = S(n);

Console.WriteLine($"S{n} kết quả là: {kq}");

Console.WriteLine();

break;

}

case 8:

{

Console.WriteLine(" Nhập số nguyên dương n: ");

int n = int.Parse(Console.ReadLine());

double kq = tichPhanSo(n);

Console.WriteLine($"S{n} kết quả là: {kq}");

Console.WriteLine();

break;

}

}

} while (LuaChon != 0);

}

static ulong tinhGiaiThua ( ulong n)

{

if (n == 0)

return 1;

else

return n \* tinhGiaiThua(n - 1);

}

static int LuyThua ( int a, int n )

{

if (n == 0)

return 1;

else

return a \* LuyThua(a, n - 1);

}

static int USCLN ( int a, int b)

{

if (b == 0)

return a;

else if (a > b)

return USCLN(a - b, b);

else

return USCLN(b - a, a);

}

static string doiNhiPhan ( int n )

{

if (n == 0)

return " ";

else

return doiNhiPhan(n / 2) + (n % 2);

}

static int soDoiTho ( int n)

{

if (n == 1 || n == 2)

return 1;

else

return soDoiTho(n - 1) + soDoiTho(n - 2);

}

static int tongS(int n)

{

if (n == 0)

return 1;

else

return (2 \* n + 1) + tongS(n - 1);

}

static int S ( int n)

{

if (n == 1)

return 1;

else

return (int)Math.Pow(-1, n + 1) \* n + S(n - 1);

}

static double tichPhanSo ( int n, int denominator = 1)

{

if (n == 1)

return 1;

else

return tichPhanSo(n - 1, denominator \* (2 \* n)) + 1 / denominator;

}

}

}