//1. Кількість грошей у дитиини 0. Хоче телефон за 10000

// let money = 0 //кількість грошей = 0

// do { //роби (повторюй)

// let fathersMoney = parseInt(prompt(`

// Я маю ${money}. Мені потрібно ${10000-money}

// Тату дай грошей`)) //кількість гроший від тата вводимо

// money += fathersMoney //гроші тата додаємо до загальної кількості грошей на телефон

// } while (money < 10000) //поки (кількість грошей <10000)

// alert("OK")

//2. Дитина заколядувала певну кількість (ввести). Хоче телефон за 10000

let money = 0 // parseInt(prompt("Введіть наявну кількість грошей")) //кількість грошей вводимо

while (money < 10000){ //поки (кількість грошей <10000)

let fathersMoney = parseInt(prompt(`

Я маю ${money}. Мені потрібно ${10000-money}

Тату дай грошей`)) //кількість гроший від тата вводимо

money += fathersMoney //гроші тата додаємо до загальної кількості грошей на телефон

}

alert("OK")

Приклад. Вводити пароль (правильний-123) поки не буде введено правильний, або не буде зроблено 3 спроби

Кількість спроб=3

Do{

Вводити пароль

Кількість спроб --

}while(спроб>0 і пароль!=123)

let attemptCount = 3 //Кількість спроб=3

do {

var password = prompt(`You have ${attemptCount} attempts. Pass`) //Вводити пароль

attemptCount-- //Кількість спроб --

} while (attemptCount > 0 && password != '123') //(спроб>0 і пароль!=123)

if (password != '123')

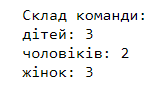
document.write("You are a thief")

else

document.write("Welcome")

Приклад. У команду поступово реєструються гравці з 3 категорій (1-Дитина. 2-Чоловік. 3-Жінка) Набір проводимо до тих пір, поки :

* членів команди буде не менше 5;
* кількість дітей не стане дорівнювати кількості чоловіків;
* кількість жінок буде більшою за кількість чоловіків.



membersCount=0 //Кільк.членів команди =0

childrenCount=0 //Кількість дітей =0

menCount=0 //Кількість чоловіків=0

womenCount=0 Кількість жінок =0

Do{

Вводимо категорію гравця

Залежно від категорії

1 : збільшуємо кількість дітей

2 : збільшуємо кількість чоловіків

3 : збільшуємо кількість жінок

Кількість членів команди ++

}while ( membersCount<5 || ( childrenCount!=menCount && womenCount<=menCount ) )

let membersCount = 0 //Кільк.членів команди =0

let childrenCount = 0 //Кількість дітей =0

let menCount = 0 //Кількість чоловіків=0

let womenCount = 0 //Кількість жінок =0

do {

let category = parseInt(prompt("1-Дитина. 2-Чоловік. 3-Жінка")) //Вводимо категорію гравця

switch (category) { //Залежно від категорії

case 1:

childrenCount++ // 1 : збільшуємо кількість дітей

break

case 2:

menCount++ // 2 : збільшуємо кількість чоловіків

break

case 3:

womenCount++ // 3 : збільшуємо кількість жінок

break

}

membersCount++ //Кількість членів команди ++

} while (membersCount < 5 || (childrenCount != menCount && womenCount <= menCount))

document.write(`

<pre>

Склад команди:

дітей: ${childrenCount}

чоловіків: ${menCount}

жінок: ${womenCount}

</pre>

`)

Приклад. Комп’ютер поступово генерує випадкове число від 1 до 5. Користувачу даємо можливість вводити відповіді поки користувач не вгадає хоча би один раз. Підрахувати кількість помилкових спроб.

Приклад. Гра «Чорні скриньки» Комп’ютер поступово генерує 3 числа, серед яких лише одне число від -100 до 100, а інші 0. Користувач вказує номер скриньки. При цьому, якщо не вгадав, то кількість спроб вибрати скриньку зменшується, а якщо вгадав, то кількість спроб не зменшується і сума виграшу збільшується (або зменшується, якщо число від’ємне) на величину, що зберігається у скриньці.

Приклад. Гра «Сірники». Комп’ютер генерує випадкову кількість сірників від 5 до 10. Двоє користувачів поступово вказують кількість сірників, які хочуть забрати (від 1 до 3). Програє той, хто забирає останній сірник.

Приклад. Гра «Маршрут з перепонами». Двоє користувачів підкидують два гральні кубики.

При цьому:

* якщо кількість балів першого кубика більша за кількість балів на другому кубику, то користувач робить кількість кроків вперед, що дорівнює різниці (кількість балів першого кубика відняти кількість балів другого кубика);
* якщо кількість балів однакова, то він робить 6 кроків вперед;
* в іншому випадку користувач стоїть на місті.

Якщо користувач зробив кроки вперед і його зроблена кількість кроків кратна 5, то він робить 10 кроків назад.

Виводити історію гри та поточну кількість кроків кожного із гравців.