

Go Program: Life Data Based Password Generator

Below is the complete Go code:

```
package main

import (
    ■"bufio"
    ■"crypto/sha256"
    ■"fmt"
    ■"os"
    ■"strconv"
    ■"strings"
)

// Charset: characters used for password generation
var charset = []rune("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$$%&*()-_+=[]{}<>?")

// function to get input from user
func ask(prompt string) string {
    ■fmt.Print(prompt + ": ")
    ■reader := bufio.NewReader(os.Stdin)
    ■input, _ := reader.ReadString('\n')
    ■return strings.TrimSpace(input)
}

// convert bytes into a password string
func bytesToPassword(hash []byte, length int) string {
    ■var pw []rune
    ■csLen := len(charset)
    ■for i := 0; i < length; i++ {
        ■idx := int(hash[i%len(hash)]) % csLen
        ■pw = append(pw, charset[idx])
    }
    ■return string(pw)
}

// hash function with iterations
func hashWithIterations(input string, iterations int) []byte {
    ■data := []byte(input)
    ■for i := 0; i < iterations; i++ {
        ■hash := sha256.Sum256(data)
        ■data = hash[:]
    }
    ■return data
}

func main() {
    ■fmt.Println("=== Detailed Life Data Based Password Generator ===")
    ■fmt.Println("Answer the questions (you can leave blank). Same answers will always generate the same pas")
    ■fmt.Println("-----")

    ■// --- Personal Information ---
    ■fullName := ask("Full Name")
    ■birthDate := ask("Birth Date (DD-MM-YYYY)")
    ■birthTime := ask("Birth Time (HH:MM)")
    ■birthCity := ask("Birth City")
    ■idLast4 := ask("ID/Passport Last 4 Digits")
    ■phoneLast4 := ask("Phone Number Last 4 Digits")

    ■// --- Physical Information ---
    ■shoeSize := ask("Shoe Size")
    ■height := ask("Height (cm)")
    ■weight := ask("Weight (kg)")
    ■siblingCount := ask("Number of Siblings")
    ■childCount := ask("Number of Children")
    ■eyeColor := ask("Eye Color (word instead of number)")

    ■// --- Education & Career ---
    ■gradYear := ask("Graduation Year")
    ■schoolNumber := ask("School Number")
}
```

```

■universityYear := ask("University Entry Year")
■employeeID := ask("Employee/Student ID")
■firstJobYear := ask("First Job Year")
■department := ask("Department/Field (e.g. Computer Engineering)")

■// --- Financial ---
■cardLast4 := ask("Bank Card Last 4 Digits")
■ibanLast3 := ask("IBAN Last 3 Digits")
■firstSalaryYear := ask("Year of First Salary")
■firstSalary := ask("First Salary Amount")
■favoriteNumber := ask("Favorite Number")

■// --- Technology & Daily Life ---
■carPlate := ask("Car Plate Code")
■postalCode := ask("Postal Code")
■firstHouseNo := ask("First House Address Number")
■marriageYear := ask("Marriage Year")
■pcRAM := ask("Computer RAM (GB)")
■phoneYear := ask("Phone Model Year")
■osVersion := ask("Operating System Version")

■// --- Extra ---
■petName := ask("Pet Name")
■favColor := ask("Favorite Color")
■luckyWord := ask("Lucky Word")
■customInput := ask("Extra Custom Data (e.g. project name, memory, etc.)")

■// --- Password settings ---
■pwLenStr := ask("Password Length (e.g. 20)")
■iterStr := ask("Number of Iterations (e.g. 50000)")

■pwLen := 20
■if pwLenStr != "" {
■    if v, err := strconv.Atoi(pwLenStr); err == nil && v > 0 {
■        pwLen = v
■    }
■}
■iterations := 50000
■if iterStr != "" {
■    if v, err := strconv.Atoi(iterStr); err == nil && v > 0 {
■        iterations = v
■    }
■}

■// --- Combine all inputs ---
■allInputs := fullName + birthDate + birthTime + birthCity + idLast4 + phoneLast4 +
■    shoeSize + height + weight + siblingCount + childCount + eyeColor +
■    gradYear + schoolNumber + universityYear + employeeID + firstJobYear + department +
■    cardLast4 + ibanLast3 + firstSalaryYear + firstSalary + favoriteNumber +
■    carPlate + postalCode + firstHouseNo + marriageYear + pcRAM + phoneYear + osVersion +
■    petName + favColor + luckyWord + customInput

■// --- Hashing with iterations ---
■finalHash := hashWithIterations(allInputs, iterations)

■// --- Generate password ---
■password := bytesToPassword(finalHash, pwLen)

■fmt.Println("-----")
■fmt.Println("Generated Password: ", password)
■fmt.Println("Note: Same answers + same settings will always generate the same password.")
■fmt.Printf("Iterations: %d | Length: %d\n", iterations, pwLen)
}

```