```
; Name            : hash.asm
;
; Build           : aclocal && autoconf && automake --add-missing --foreign
;                   mkdir build
;                   cd build
;                   ../configure
;                   make
;
; Description     : an example of glib 2.0 hash tables
;
; Source:         : https://github.com/steshaw/gtk-examples

bits 64

[list -]
    extern    g_hash_table_new
    extern    g_hash_table_destroy
    extern    g_hash_table_insert
    extern    g_hash_table_foreach
    extern    g_direct_hash
    extern    g_direct_equal
    extern    g_print
    extern    exit
[list +]

section .rodata

    prtKeyValue:      db    "%d : %s %s %s",10,0
    testline1:        db    "-----",10,0
    testline2:        db    "compare %s - %s",10,0
    testline3:        db    "create hash for %10s : %d",10,0
    userdata:         db    "=>",0

    keys:
    .fred:            db    "Fred",0
    .mary:            db    "Mary",0
    .sue:             db    "Sue",0
    .john:            db    "John",0
    .shelley:         db    "Shelley",0
    .markus:          db    "Markus",0
    .renato:          db    "Renato",0
    values:
    .boring:          db    "Boring",0
    .shifty:          db    "Shifty",0
    .nice:            db    "Nice",0
    .strange:         db    "Strange",0
    .abnormal:        db    "Abnormal",0
    .absentminded:    db    "Absent minded",0
    .paranoid:        db    "Paranoid",0
    .smart:           db    "Smart",0
    .intelligent:     db    "Intelligent",0
    .stubbytoes:      db    "Stubby toes",0
    datapairs:        dq    keys.fred,values.boring
                      dq    keys.mary,values.shifty
                      dq    keys.sue,values.nice
                      dq    keys.john,values.strange
                      dq    keys.shelley,values.abnormal
                      dq    keys.markus,values.absentminded
                      dq    keys.renato,values.paranoid
                      dq    keys.renato,values.smart
                      dq    keys.renato,values.intelligent
                      dq    keys.renato,values.stubbytoes
                      dq    0

section .data

    hTable:           dq    0

section .text
global _start

_start:
    ;create new hashtable and store handler in hTable
    mov     rsi,g_direct_equal
    mov     rdi,g_direct_hash
    call    g_hash_table_new
    mov     [hTable],rax
    ;read all key/value pairs and store them in the hashtable
    mov     r15,rax
    mov     r14,datapairs
.repeat:
    mov     rax,[r14]
    test    rax,rax
    jz      .endoflist
    mov     rdx,[r14+8]
    mov     rsi,[r14]
    mov     rdi,r15
    call    g_hash_table_insert
    add     r14,16
    jmp     .repeat
.endoflist:
    ;print the hashtable items
    mov     rdi,[hTable]
    mov     rsi,fnPrintTable
    mov     rdx,userdata
    call    g_hash_table_foreach
    ;destroy the hashtable
    mov     rdi,[hTable]
    call    g_hash_table_destroy
    ;exit the program
    xor     rdi,rdi
    call    exit

fnPrintTable:
    ;routine to print a single hashtable key/value pair
    ;with mask "%d : %s %s %s"
    ;rdi = output mask
    ;rsi = pointer to key
    ;rcx = pointer to value
    ;rdx = pointer to user data (in this case =>)
    ;%d(rsi) %s(rsi) %s(rdx) %s(rcx)
    ;the function starts with rdi=pointer to key
    ;                         rsi=pointer to value
    ;                         rdx=pointer tu user data
    push    rbp
    mov     rbp,rsp
    mov     r8,rsi                  ;pointer to value
    mov     rcx,rdx                 ;rcx = user data (=>)
    mov     rdx,rdi                 ;rdx = pointer to key displayed as string
    mov     rsi,rdi                 ;rsi = pointer to key displayed as pointer
    mov     rdi,prtKeyValue         ;rdi = mask
    xor     rax,rax
    call    g_print
    mov     rsp,rbp
    pop     rbp
    ret
```