

BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 10

Họ và tên: Nguyễn Quý Đức

MSSV: 20235682

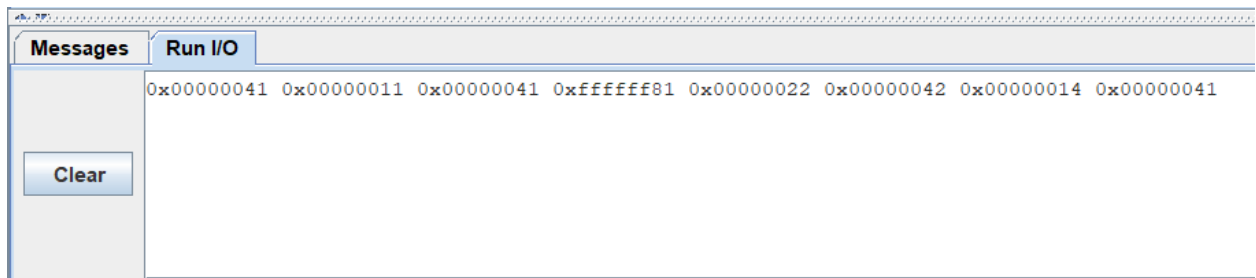
Mã HP: IT3280

Mã lớp: 156788

1. Assignment

- Với mỗi dòng (row), nó:
 - Gửi mã dòng (t3) vào địa chỉ IN_ADDRESS_HEXa_KEYBOARD.
 - Đọc mã phím từ địa chỉ OUT_ADDRESS_HEXa_KEYBOARD.
 - Nếu có phím được nhấn (giá trị đọc $\neq 0$):
 - In mã phím ra màn hình dưới dạng hexadecimal.
 - Ngủ 100ms để tránh đọc lặp nhanh.
 - Chuyển sang dòng tiếp theo (dịch trái 1 bit).
- Sau khi quét hết 4 dòng (0x01, 0x02, 0x04, 0x08), chương trình quay lại quét tiếp.

Kết quả khi bấm lần lượt từng nút 2-0-2-3-5-6-8-2:



```
.eqv IN_ADDRESS_HEXa_KEYBOARD 0xFFFF0012
# Receive row and column of the key pressed, 0 if not key pressed
# Eg. equal 0x11, means that key button 0 pressed.
# Eg. equal 0x28, means that key button D pressed.
.eqv OUT_ADDRESS_HEXa_KEYBOARD 0xFFFF0014
.eqv newline 0xa
.text
main:
    li t1, IN_ADDRESS_HEXa_KEYBOARD
    li t2, OUT_ADDRESS_HEXa_KEYBOARD

polling:
    li t3, 0x01 # start with row 1
```

row_loop:

```
sb t3, 0(t1) # must reassign expected row
lb a0, 0(t2) # read scan code of key button
beqz a0, next_row # if no key pressed, continue to next row
```

print:

```
li a7, 34 # print integer (hexa)
ecall
li a0, newline
li a7, 11
ecall
```

sleep:

```
li a0, 100 # sleep 100ms
li a7, 32
ecall
```

next_row:

```
slli t3, t3, 1 # move to next row (shift left 1 bit)
li t4, 0x10 # maximum row mask
blt t3, t4, row_loop # if not finished all rows, continue row_loop
j polling # continue polling
```

2. Assignment 2

The screenshot displays a debugger interface with two main panels. The left panel, titled 'Labels', shows a list of labels and their corresponding memory addresses for a file named 'w11h2.asm'. The right panel, titled 'Registers', shows a table of registers, their numbers, and their current values.

Label	Address
main	0x00400000
loop	0x00400028
end_main	0x00400038
handler	0x00400038
message	0x10010000

Name	Number	Value
ustatus	0	0x00000010
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000100
utvec	5	0x00400038
uscratch	64	0x00000000
uepc	65	0x00400034
ucause	66	0x80000008
utval	67	0x00000200
uip	68	0x00000000
cycle	3072	0x0000001f
time	3073	0x79f4d619
instret	3074	0x0000001f
cycleh	3200	0x00000000
timeh	3201	0x00000196
instreth	3202	0x00000000

- Trong hình trên là trạng thái các thanh ghi khi xảy ra ngắt do bấm phím ở keypad:
 - ustatus = 0x10 nghĩa là: đang trong lúc xử lý ngắt, chương trình khóa tạm ngắt (0th bit UIE = 0), lưu lại trạng thái trước đó interrupt đã được bật (4th bit UPIE = 1).
 - uie = 0x00000100 nghĩa là: chỉ cho phép ngắt từ thiết bị ngoại vi (8th bit UEIE = 1 User External Interrupt Enable và các bit khác = 0)
 - utvec = 0x00400038 lưu địa chỉ của chương trình con interruption handler “handler”
 - uepc = 0x00400034 lưu giá trị thanh ghi pc khi xảy ra lệnh ngắt
 - ucause = 0x80000008 nghĩa là đang xảy ra interrupt (31st bit = 1) với nguyên nhân là external interrupt (bit[30:0] EXCCODE = 8)

Registers			Floating Point			Control and Status		
Name	Number	Value						
ustatus	0	0x00000001						
fflags	1	0x00000000						
frm	2	0x00000000						
fcsr	3	0x00000000						
uie	4	0x00000100						
utvec	5	0x00400038						
uscratch	64	0x00000000						
uepc	65	0x00000000						
ucause	66	0x00000000						
utval	67	0x00000000						
uip	68	0x00000000						
cycle	3072	0x0000009a						
time	3073	0x7a28e80f						
instret	3074	0x0000009a						
cycleh	3200	0x00000000						
timeh	3201	0x00000196						
instreth	3202	0x00000000						

- Trong hình trên là trạng thái chưa xảy ra interrupt:
 - ustatus = 0x1 nghĩa là: chưa interrupt, cho phép ngắt xảy ra (0th bit UIE = 1)
 - uie = 0x00000100
 - utvec = 0x00400038
 - uepc = 0x00000000 do chưa xảy ra ngắt nên không lưu địa chỉ xảy ra ngắt
 - ucause = 0x00000000 tương tự như trên, nguyên nhân ngắt = 0

3. Assignment 3

Registers		Floating Point	Control and Status
Name	Number	Value	
ustatus	0	0x00000010	
fflags	1	0x00000000	
frm	2	0x00000000	
fcscr	3	0x00000000	
uie	4	0x00000100	
utvec	5	0x0040005c	
uscratch	64	0x00000000	
uepc	65	0x00400050	
ucause	66	0x80000008	
utval	67	0x00000200	
uip	68	0x00000000	
cycle	3072	0x00000087	
time	3073	0x7ad1299c	
instret	3074	0x00000087	
cycleh	3200	0x00000000	
timeh	3201	0x00000196	
instreth	3202	0x00000000	

- Trong hình trên là trạng thái các thanh ghi khi xảy ra ngắt do bấm phím ở keypad:
 - ustatus = 0x10 nghĩa là: đang trong lúc xử lý ngắt, chương trình khóa tạm ngắt (bit 0 UIE = 0), lưu lại trạng thái trước đó interrupt đã được bật (bit 4 UIPE = 1)
 - uie = 0x00000100 nghĩa là: chỉ cho phép ngắt từ thiết bị ngoại vi (bit 8 UEIE = 1 User External Interrupt Enable và các bit khác = 0)
 - utvec = 0x00400070 lưu địa chỉ của chương trình con interruption handler "handler"
 - uepc = 0x00400050 lưu giá trị thanh ghi pc khi xảy ra lệnh ngắt
 - ucause = 0x80000008 nghĩa là đang xảy ra interrupt (bit 31 = 1), với nguyên nhân là external interrupt (EXCCODE[30:0] = 8)

```
.eqv IN_ADDRESS_HEXKEYBOARD 0xFFFF0012
```

```
.eqv OUT_ADDRESS_HEXKEYBOARD 0xFFFF0014
```

```
.data
```

```
message: .asciz "\nKey scan code: "
```

```
newline: .asciz "\n"
```

```
# -----
```

```
# MAIN Procedure
```

```
# -----
```

```
.text
```

```
main:
```

```
# Load the interrupt service routine address to the UTVEC register
```

```

        la t0, handler
        csrrs zero, utvec, t0
# Set the UEIE (User External Interrupt Enable) bit in UIE register
        li t1, 0x100
        csrrs zero, uie, t1 # uie - ueie bit (bit 8)
# Set the UIE (User Interrupt Enable) bit in USTATUS register
        csrrsi zero, ustatus, 1 # ustatus - enable uie (bit 0)
# Enable the interrupt of keypad of Digital Lab Sim
        li t1, IN_ADDRESS_HEXA_KEYBOARD
        li t3, 0x80 # bit 7 = 1 to enable interrupt
        sb t3, 0(t1)

# -----
# Loop to print a sequence numbers
# -----

        xor s0, s0, s0 # count = s0 = 0
loop:
        addi s0, s0, 1 # count = count + 1
prn_seq:
        addi a7, zero, 1
        add a0, s0, zero # Print auto sequence number
        ecall
# Print EOL
        addi a7, zero, 4
        la a0, newline
        ecall
sleep:
        addi a7, zero, 32
        li a0, 300 # Sleep 300 ms
        ecall
        j loop
end_main:

# -----
# Interrupt service routine

```

```

# -----
handler:
# Saves the context
    addi sp, sp, -16
    sw a0, 0(sp)
    sw a7, 4(sp)
    sw t1, 8(sp)
    sw t2, 12(sp)

# Handles the interrupt
prn_msg:
    addi a7, zero, 4
    la a0, message
    ecall

get_key_code:
    li t1, IN_ADDRESS_HEXKEYBOARD
    li t2, OUT_ADDRESS_HEXKEYBOARD

polling:
    li t3, 0x01 # start with row 1
row_loop:
    sb t3, 0(t1) # must reassign expected row
    lb a0, 0(t2) # read scan code of key button
    beqz a0, next_row # if no key pressed, continue to next row

print:
    li a7, 34 # print integer (hexa)
    ecall
    li a0, 0xa
    li a7, 11
    ecall

next_row:
    slli t3, t3, 1 # move to next row (shift left 1 bit)

```

```

        li t4, 0x10 # maximum row mask

        blt t3, t4, row_loop # if not finished all rows, continue row_loop

# After polling, must re-enable keypad interrupt
re_enable_interrupt:
        li t1, IN_ADDRESS_HEXKEYBOARD

        li t3, 0x80 # bit 7 = 1 to enable interrupt again

        sb t3, 0(t1)

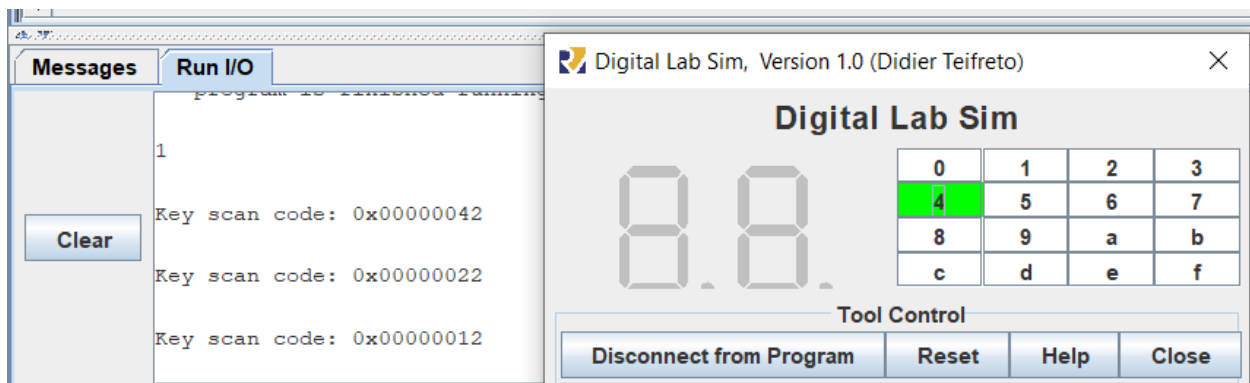
        j restore

restore:
# Restores the context
        lw t2, 12(sp)
        lw t1, 8(sp)
        lw a7, 4(sp)
        lw a0, 0(sp)
        addi sp, sp, 16

# Back to the main procedure
        Uret

```

Console hiển thị khi bấm nút 6-5-4 như sau:



4. Assignment 4

Registers Floating Point Control and Status		
Name	Number	Value
ustatus	0	0x00000010
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000110
utvec	5	0x0040004c
uscratch	64	0x00000000
uepc	65	0x00400044
ucause	66	0x80000004
utval	67	0x00000010
uip	68	0x00000000
cycle	3072	0x0000169c
time	3073	0x7b0ce26d
instret	3074	0x0000169c
cycleh	3200	0x00000000
timeh	3201	0x00000196
instreth	3202	0x00000000

- Trong hình trên là trạng thái các thanh ghi khi xảy ra ngắt do Timer đã đếm được 1000ms:
 - ustatus = 0x00000010 nghĩa là: đang trong lúc xử lý ngắt, chương trình tạm khóa global interrupt (bit 0 UIE = 0), nhưng lưu lại trạng thái trước đó interrupt đã bật (bit 4 UIPE = 1)
 - uie = 0x00000110 nghĩa là: chương trình cho phép cùng lúc 2 loại interrupt:
 - Bit 8 (UEIE) = 1 → External interrupt enable (cho phép ngắt từ thiết bị ngoại vi như Keypad)
 - Bit 4 (UTIE) = 1 → Timer interrupt enable (cho phép ngắt từ Timer)
 - utvec = 0x0040004c lưu địa chỉ của chương trình con xử lý ngắt (handler), đúng theo label "handler" bắt đầu từ địa chỉ 0x0040004c
 - uepc = 0x00400044 lưu giá trị PC tại thời điểm xảy ra ngắt, nghĩa là lúc đó chương trình đang thực thi tại địa chỉ 0x00400044
 - ucause = 0x80000004 nghĩa là: đang có interrupt xảy ra (bit 31 = 1), với nguyên nhân là Timer interrupt (EXCCODE[30:0] = 4)

Registers Floating Point Control and Status		
Name	Number	Value
ustatus	0	0x00000010
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000110
utvec	5	0x0040004c
uscratch	64	0x00000000
uepc	65	0x00400044
ucause	66	0x80000008
utval	67	0x00000200
uip	68	0x00000000
cycle	3072	0x000001f2
time	3073	0x7b035638
instret	3074	0x000001f2
cycleh	3200	0x00000000
timeh	3201	0x00000196
instreth	3202	0x00000000

- Trong hình trên là trạng thái các thanh ghi khi xảy ra ngắt do bấm phím ở keypad:
 - ustatus = 0x00000010 nghĩa là: đang trong lúc xử lý ngắt, chương trình tạm khóa global interrupt (bit 0 UIE = 0), đồng thời lưu trạng thái trước đó interrupt đã bật (bit 4 UPIE = 1)
 - uie = 0x00000110 nghĩa là: chương trình cho phép cùng lúc 2 loại interrupt:
 - Bit 8 (UEIE) = 1 → Cho phép External interrupt (thiết bị ngoại vi như Keypad)
 - Bit 4 (UTIE) = 1 → Cho phép Timer interrupt
 - utvec = 0x0040004c lưu địa chỉ của chương trình con xử lý ngắt (handler), khớp với Label "handler" ở địa chỉ 0x0040004c
 - uepc = 0x00400044 lưu giá trị PC tại thời điểm xảy ra ngắt, tức là lúc đó chương trình chính đang thực thi tại địa chỉ 0x00400044
 - ucause = 0x80000008 nghĩa là: có interrupt xảy ra (bit 31 = 1), và nguyên nhân là External interrupt (EXCCODE[30:0] = 8) — tương ứng với ngắt do thiết bị ngoại vi Keypad

Registers Floating Point Control and Status		
Name	Number	Value
ustatus	0	0x00000001
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000110
utvec	5	0x0040004c
uscratch	64	0x00000000
uepc	65	0x00400048
ucause	66	0x80000008
utval	67	0x00000200
uip	68	0x00000000
cycle	3072	0x00000742
time	3073	0x7b07dce4
instret	3074	0x00000742
cycleh	3200	0x00000000
timeh	3201	0x00000196
instreth	3202	0x00000000

- Trong hình trên là trạng thái các thanh ghi khi bình thường, không bị interrupt:
 - ustatus = 0x1 nghĩa là: chưa interrupt, cho phép ngắt xảy ra (0th bit UIE = 1)
 - uie = 0x00000110 nghĩa là: chương trình cho phép cùng lúc 2 loại interrupt:
 - Bit 8 (UEIE) = 1 → Cho phép External interrupt (thiết bị ngoại vi như Keypad)
 - Bit 4 (UTIE) = 1 → Cho phép Timer interrupt

5. Assignment 5

Registers	Floating Point	Control and Status
Name	Number	Value
ustatus	0	0x00000000
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000000
utvec	5	0x0040001c
uscratch	64	0x00000000
uepc	65	0x00000000
ucause	66	0x00000000
utval	67	0x00000000
uip	68	0x00000000
cycle	3072	0x00000003
time	3073	0x7b1ba586
instret	3074	0x00000003
cycleh	3200	0x00000000
timeh	3201	0x000000196
instreth	3202	0x00000000

Sau lệnh csrrsi zero, ustatus, 1:

- ustatus = 0x00000000 nghĩa là không bật global interruption
- uie = 0x00000000 nghĩa là: chương trình không bật cho phép bất kỳ interrupt nào lúc này (không bật UEIE, UTIE,...)
- utvec = 0x0040001c lưu địa chỉ của chương trình con xử lý exception (handler hoặc tương tự), đang đặt tại địa chỉ 0x0040001c
- uepc = 0x00000000 do chưa xảy ra exception
- ucause = 0x00000000 do chưa xảy ra exception

Registers	Floating Point	Control and Status
Name	Number	Value
ustatus	0	0x00000010
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000000
utvec	5	0x0040001c
uscratch	64	0x00000000
uepc	65	0x00400010
ucause	66	0x00000005
utval	67	0x00000000
uip	68	0x00000000
cycle	3072	0x00000004
time	3073	0x7b156655
instret	3074	0x00000004
cycleh	3200	0x00000000
timeh	3201	0x00000196
instreth	3202	0x00000000

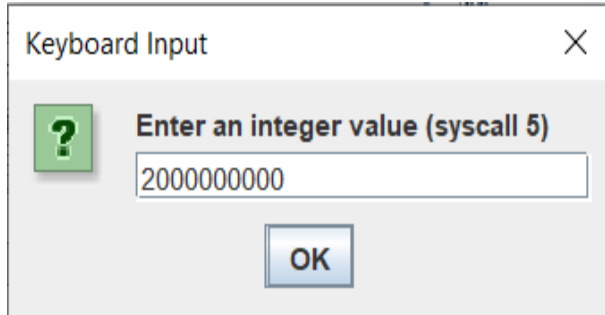
Sau lệnh lw zero, 0

- ustatus = 0x00000010 nghĩa là: chương trình đang trong quá trình xử lý trap/exception, global interrupt bị tạm khóa (bit 0 UIE = 0), nhưng lưu trạng thái trước đó interrupt đã bật (bit 4 UIE = 1)
- uie = 0x00000000 nghĩa là: chương trình không bật cho phép bất kỳ interrupt nào lúc này (không bật UEIE, UTIE,)
- utvec = 0x0040001c lưu địa chỉ của chương trình con xử lý exception (handler hoặc tương tự), đang đặt tại địa chỉ 0x0040001c
- uepc = 0x00400010 lưu địa chỉ PC khi xảy ra exception, tức là chương trình đã bị trap khi đang thực thi lệnh tại địa chỉ 0x00400010
- ucause = 0x00000005 nghĩa là:
 - Bit 31 = 0 → Đây là exception (không phải interrupt)
 - EXCCODE[30:0] = 5 → Load Access Fault: lỗi truy cập khi thực hiện lệnh load bộ nhớ
- utval = 0x00000010 chứa địa chỉ bộ nhớ gây ra lỗi (ví dụ truy cập vào địa chỉ 0x10 gây lỗi)

6. Assignment 6

Chương trình tự phát hiện overflow → set bit USIP → CPU thấy có pending software interrupt → nhảy vào handler để xử lý exception.

- Khi người dùng nhập 2 số nguyên có tổng lớn hơn $\text{INT32_MAX} = 2147483647$

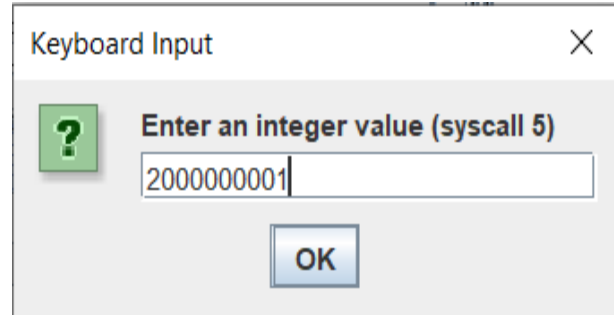


Keyboard Input

Enter an integer value (syscall 5)

2000000000

OK



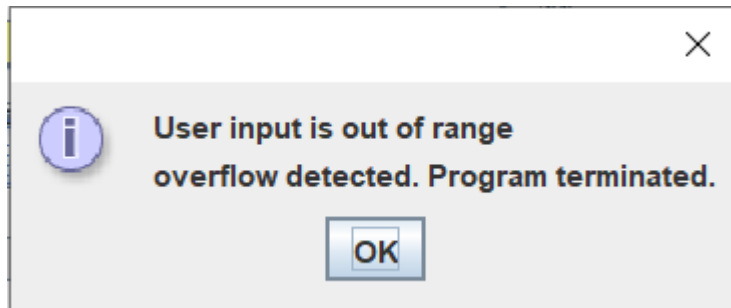
Keyboard Input

Enter an integer value (syscall 5)

2000000001

OK

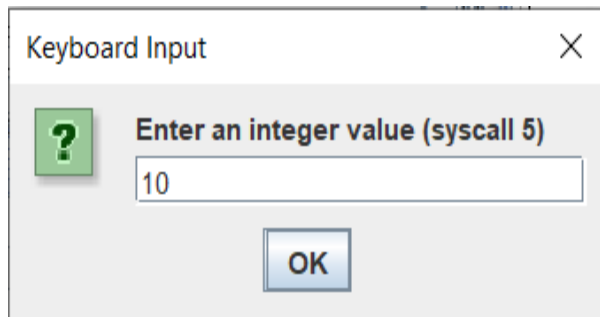
Sẽ có thông báo tràn số:



User input is out of range
overflow detected. Program terminated.

OK

- Khi người dùng nhập 2 số mà tổng không bị tràn:

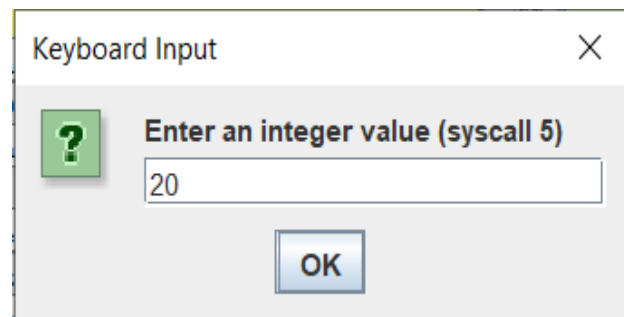


Keyboard Input

Enter an integer value (syscall 5)

10

OK



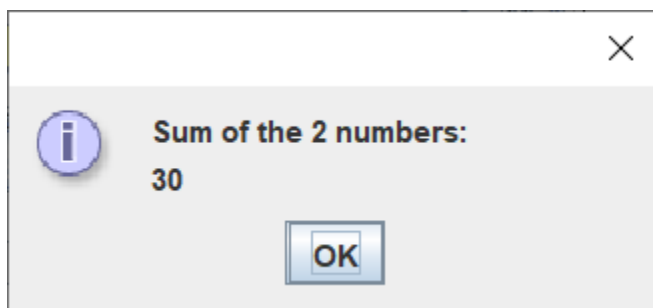
Keyboard Input

Enter an integer value (syscall 5)

20

OK

Sẽ in ra thông báo tổng 2 số:



Sum of the 2 numbers:
30

OK