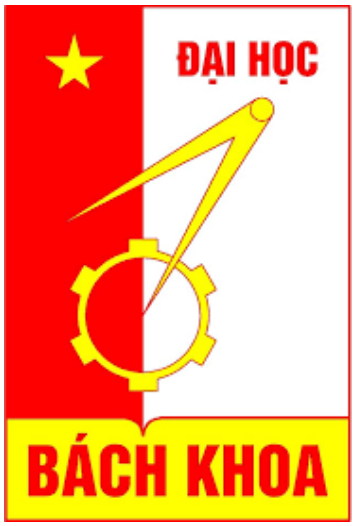
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công Nghệ Thông Tin và Truyền Thông

==========



THỰC HÀNH
KIẾN TRÚC MÁY TÍNH

Nhóm: **8**
Sinh viên: **Hồ Khánh Dương - 20176736**
Đỗ Thị Hồng Thảo -20176878
Mã lớp: **113835**
Giáo viên hướng dẫn: **Lê Xuân Thành**

Năm học: 2019-2020

MỤC LỤC

ĐỀ BÀI

Project 15.

You are given an array of integers. On each move you are allowed to increase exactly one of its elements by one. Find the minimal number of moves required to obtain a strictly increasing sequence from the input.
Example: For inputArray = [1, 1, 1], the output should be arrayChange(inputArray) =3. The minimal number of moves needed to obtain a strictly increasing sequence from inputArray. It's guaranteed that for the given test cases the answer always fits signed32-bit integer type.

Project 16.

Given a sequence of integers as an array, determine whether it is possible to obtain a strictly increasing sequence by removing no more than one element from the array.
Note: sequence a0, a1, ..., an is considered to be strictly increasing if a0 < a1 < ... <an. Sequences containing only one element are also considered to be strictly increasing.
Example:
-
For sequence =[1,3,2,1],the out put should be almostIncreasingSequence(sequence) = false. There is no one element in this array that can be removed in order to get a strictly increasing sequence.
- For sequence = [1, 3, 2], the output should be almostIncreasingSequence(sequence)= true. You can remove 3 from the array to get the strictly increasing sequence [1, 2]. Alternately, you can remove 2 to get the strictly increasing sequence [1, 3].

Project 21.

Let's define the digit degree of some positive integer as the number of times we need to replace this number with the sum of its digits until we get to a one digit number.
Given an integer, find its digit degree.
Example
- For n = 5 , the output should be digitDegree(n) = 0 ;
- For n = 100 , the output should be digitDegree(n) = 1 . 1 + 0 + 0 = 1 .
- For n = 91 , the output should be digitDegree(n) = 2 . 9 + 1 = 10 -> 1 + 0 = 1 .

Project 24.

Cyclone Word (challenge)
Cyclone words are English words that have a sequence of characters in alphabetical order when following a cyclic pattern.
Example:
Write a function to determine whether a word passed into a function is a cyclone word. You can assume that the word is made of only alphabetic characters, and is separated by whitespace.
is_cyclone_phrase("adjourned") # => True
is_cyclone_phrase("settled") # => False

Ý TƯỞNG THUẬT TOÁN

Project 15.

Cho nhập số phần tử (n) và các phần tử trong mảng (A), kiểm tra các số nhập vào có phải số nguyên không.

Cho biến i chạy từ 1 đến n, so sánh A[i] và A[i-1].

Nếu A[i-1] < A[i] thì lặp lại

Nếu không : count = A[i] - A[i-1] +1

A[i] += count

sum += count

count = 0

Trong đó : count : số lần cần tăng 1 đơn vị vào A[i] để A[i-1] < A[i]

sum : tổng số lần cần tăng 1 đơn vị để được mảng tăng tuyệt đối

Project 16.

So sánh A[0] với A[1] => Nếu A[0]>=A[1] =>count++

Cho i chạy từ 0 đến length(A)-1

So sánh A[i] với A[i+1]

If A[i]>=A[i+1]

=> count++

=> So sánh tiếp A[i-1] với A[i+1]

Nếu A[i-1]>=A[i+1] =>false

Nếu không =>Tiếp tục

i++

Nếu count=2 =>false

Nếu count <2 =>true

Project 21.

Nếu A > 10

count++

Nếu A > 0

sumA += A%10

A = A/10

Lặp lại đến khi A < 0

A = sumA

Lặp lại đến khi A < 10

Project 24.

Lưu các ký tự của từ vào 1 mảng A

Cho con trỏ i chạy từ phải qua trái, j chạy từ trái qua phải

Nếu A[i] < A[j] thì không phải từ bão tố

Nếu i = j thì đấy là từ bão tố

SOURCE CODE

Project 15.

- *Ý nghĩa các thanh ghi được sử dụng*

  $s2 : Array[i]
  $s0: Array[j]

  $t1 : số lần cần tăng 1 đơn vị của 1 lần

  $s3 : giá trị Array[j] mỗi lần đọc từ chuỗi

  $s5 : tổng số lần cần tăng 1 đơn vị

- *Source code*

```
# You are given an array of integers. On each move you are allowed to increase exactly one of its elements by one.
# Find the minimal number of moves required to obtain a strictly increasing sequence from the input.
.data
Array: .word
Mess1:.asciiz "Nhap so phan tu"
Mess2:          .asciiz "Nhap cac phan tu phan cach nhau bang dau , "
KetQua: .asciiz "Ket Qua : "
Err:    .asciiz "Loi tai vi tri : "
Again: .asciiz "Chay lai"
string:  .space 100
DauPhay: .asciiz ",  "

.text
main:
        la $s2, Array           # gan dia chi mang cho s2
        jal Nhap
        nop


        li $s5, 0
        sub $s0, $s0, 4
        jal Tinh
        nop

        j end_main
#---------------------------------------------------------
# Nhap : de nhap day so duoc ngan cach bang dau ',' , kiem tra va cho vao mang Array
# $a1 luu tru chuoi nhap vao
# $s0 luu tru dia chi A[i]
#---------------------------------------------------------
Nhap:
        li  $v0, 54
        la  $a0, Mess2
        la  $a1, string
        la  $a2, 100
        syscall
```

```
        la $a1, string

        la $s0, Array           # gan dia chi mang cho s0

        li  $t0, 0              # i = 0  : con tro trong chuoi nhap vao

        li  $s7, 1              # co the la so am
Check:
        add $t1, $t0, $a1       # t1 tro toi vi tri String[i]
        lb  $t3, 0($t1)         # t3 = String[i]
        li  $t2, 47             # t2 = ':'
        li  $t4, 58             # t4 = '/'

        beq $t3, 10, end_check          # if(String[i] == '\n') end check
        beq $t3, 44, KTSA       # if(String[i] == ',') next
        beq $t3, 32, Next       # if(String[i] == ' ') next
        beq $t3, 45, So_Am      # Kiem tra so am

        slt $t5, $t2, $t3       # String[i] <= 9 ?
        slt $t6, $t3, $t4       # String[i] >= 0 ?
        and $t7, $t5, $t6       # 0 <= String[i] <= 9 ?
        bne $t7, 0, ADD1        # if(true) qua ADD1
        beq $t7, 0, err         # else error
ADD1:
        li  $s7, 0              # Khong the la so am
        li  $s6, 1              # Co so
        sub $t3, $t3, 1         # string[i] --
        sub $t7, $t3, $t2       # $t7 = string[i] - 47
        mul $s3, $s3, 10        # A[j] = A[j]*10
        add $s3, $s3, $t7       # A[j] += $t7
        j   Next
KTSA:
        beq $s5, 0, ADD2        # Neu khong phai so am thi them vao mang
        sub $s3, $0, $s3        # Neu la so am thi A[j] = 0-A[j]
        li  $s5, 0              # Tat co so am
ADD2:
        beq $s6, 0, err         # Neu khong co so
        sw  $s3, 0($s0)                 # them A[j] vao mang
        add $s0, $s0, 4                 # j++
        li  $s6, 0              # Khong co so
        li  $s7, 1              # Co the tiep theo la so am
        li  $s3, 0              # reset so nhap vao
        j   Next
So_Am:
        beq $s7, 0, err         # Neu khong the la so am thi bao loi
        li $s5, 1               # Bat co so am
        j  Next
Next:
        add $t0, $t0, 1         # i++
```

```
        j   Check
Chuyen:
        sub $s3, $0, $s3
        li  $s5, 0
end_check:
        beq $s6, 0, err         # Neu khong co so
        beq $s5, 1, Chuyen
        sw  $s3, 0($s0)                 # them A[j] vao mang
        add $s0, $s0, 4                 # j++
        jr  $ra                 # Quay lai main

#---------------------------------------------------------
# Tinh : de tinh xem can bao nhieu lan tang 1 don vi de dat duoc mang tang tuyet doi
#       Bang cach xet tu A[0] den A[n] va tang tung phan tu neu can thiet
#---------------------------------------------------------
Tinh:
        slt $t0, $s2 , $s0      # Dia chi A[j+1] > A[i] ?
        beq $t0, 0, end         # if(false) thi ket thuc
        add $t1, $0, $0         # reset count = 0
        lw  $t2, 0($s2)         # $t5 = A[i]
        lw  $t3, 4($s2)         # $t6 = A[i+1]

        slt $t4, $t2, $t3       # A[i] < A[i+1] ?
        beq $t4, $0, Tang       # if(false) Tang
        add $s2, $s2, 4
        j   Tinh
Tang:
        sub $t1, $t2, $t3       # count = A[i] - A[i+1]
        add $t1, $t1, 1         # count ++
        add $t3, $t3, $t1       # A[i+1] += count
        sw  $t3, 4($s2)                 # load lai A[i+1] vao mang
        add $s5, $s5, $t1       # sum += count
        add $s2, $s2, 4                 # j++
        j   Tinh
end:
        li $v0, 56
        la $a0, KetQua
        add $a1, $s5, 0
        syscall
        jr  $ra
err:
        sub $s1, $s0, $s2       # s1 = A[i] - A[0]
        li  $t0, 4
        div $s1, $t0
        mflo $t4                # t4 = s1 / 4 : vi tri nhap sai
        add $t4, $t4, 1

        li $v0, 56
        la $a0, Err
        add $a1, $t4, $0
```

```
        syscall
        j  end_main
end_main:
```

Project 16.

- *Ý nghĩa các thanh ghi được sử dụng*
  - $s2 : Array[i]
  - $s0: Array[j]
  - $s6: count
- *Source code*

```
# Given a sequence of integers as an array, determine whether it is possible to obtain a strictly increasing sequence by removing no more than one element from the array.
.data
Array: .word
Mess1:          .asciiz "Nhap so phan tu"
Mess2:          .asciiz "Nhap cac phan tu phan cach nhau bang dau , "
KetQua:         .asciiz "Ket Qua : "
Err:            .asciiz "Loi tai vi tri : "
Again:          .asciiz "Chay lai"
true:           .asciiz "true"
false:          .asciiz "false"
string:         .space 100


.text
main:
        la $s2, Array           # gan dia chi mang cho s2
        jal Nhap
        nop


        li $s5, 0
        sub $s0, $s0, 4
        jal SoSanh
        nop

        j end_main
#---------------------------------------------------------------
# Nhap : de nhap day so duoc ngan cach bang dau ',' , kiem tra va cho vao mang Array
# $a1 luu tru chuoi nhap vao
# $s0 luu tru dia chi A[i]
#---------------------------------------------------------------
Nhap:
        li  $v0, 54
        la  $a0, Mess2
        la  $a1, string
```

```
        la  $a2, 100
        syscall

        la $a1, string

        la $s0, Array           # gan dia chi mang cho s0

        li  $t0, 0              # i = 0  : con tro trong chuoi nhap vao

        li  $s7, 1              # co the la so am
Check:
        add $t1, $t0, $a1       # t1 tro toi vi tri String[i]
        lb  $t3, 0($t1)         # t3 = String[i]
        li  $t2, 47             # t2 = ':'
        li  $t4, 58             # t4 = '/'

        beq $t3, 10, end_check          # if(String[i] == '\n') end check
        beq $t3, 44, KTSA       # if(String[i] == ',') next
        beq $t3, 32, Next       # if(String[i] == ' ') next
        beq $t3, 45, So_Am      # Kiem tra so am

        slt $t5, $t2, $t3       # String[i] <= 9 ?
        slt $t6, $t3, $t4       # String[i] >= 0 ?
        and $t7, $t5, $t6       # 0 <= String[i] <= 9 ?
        bne $t7, 0, ADD1        # if(true) qua ADD1
        beq $t7, 0, err         # else error
ADD1:
        li  $s7, 0              # Khong the la so am
        li  $s6, 1              # Co so
        sub $t3, $t3, 1         # string[i] --
        sub $t7, $t3, $t2       # $t7 = string[i] - 47
        mul $s3, $s3, 10        # A[j] = A[j]*10
        add $s3, $s3, $t7       # A[j] += $t7
        j   Next
KTSA:
        beq $s5, 0, ADD2        # Neu khong phai so am thi them vao mang
        sub $s3, $0, $s3        # Neu la so am thi A[j] = 0-A[j]
        li  $s5, 0              # Tat co so am
ADD2:
        beq $s6, 0, err         # Neu khong co so
        sw  $s3, 0($s0)                 # them A[j] vao mang
        add $s0, $s0, 4                 # j++
        li  $s6, 0              # Khong co so
        li  $s7, 1              # Co the tiep theo la so am
        li  $s3, 0              # reset so nhap vao
        j   Next
So_Am:
        beq $s7, 0, err         # Neu khong the la so am thi bao loi
        li $s5, 1               # Bat co so am
        j  Next
```

```
Next:
        add $t0, $t0, 1         # i++
        j   Check
Chuyen:
        sub $s3, $0, $s3
        li  $s5, 0
end_check:
        beq $s6, 0, err         # Neu khong co so
        beq $s5, 1, Chuyen
        sw  $s3, 0($s0)                 # them A[j] vao mang
        add $s0, $s0, 4                 # j++
        jr  $ra                 # Quay lai main

#---------------------------------------------------------
# Kiem tra : Kiem tra day so nhap vao co phai la chuoi tang nghiem ngat hay khong, bang cach loai
#       bo khong qua mot phan tu khoi mang
#---------------------------------------------------------
SoSanh:
        slt     $t0, $s2 , $s0  # Dia chi A[j+1] > A[i] ?
        beq     $t0, 0, sai     # if(false) thi ket thuc tra ve false
        li      $s6,0           # count=0
        lw      $t2,0($s2)      # t2=A[0]
        lw      $t3,4($s2)      # t3=A[1]
        slt     $t5,$t2,$t3     # A[0]<A[1]             nguoc lai se la A[1]<=A[0]
        addi    $s2,$s2,4
        bne     $t5,$zero,for   #neu t5=0
        addi    $s6,$s6,1       #count++

for:
        slt     $t0, $s2 , $s0  # Dia chi A[j+1] > A[i] ?
        beq     $t0, 0, dung    # if(false) thi ket thuc tra ve true
        lw      $t3,0($s2)      # t3=A[i]
        lw      $t4,4($s2)      # t4=A[i+1]
        slt     $t5,$t3,$t4     # A[i]<A[i+1]           nguoc lai se la A[i+1]<=A[i]
        addi    $s2,$s2,4

        bne     $t5,$zero,for   # neu t5=0 thi quay lai for
        addi    $s6,$s6,1       # count++
        addi    $t7,$0,1        # t7=1
        slt     $t6,$t7,$s6     # so sanh count voi 1
        bne     $t6,$zero,sai   # neu count>1 => sai

        slt     $t0, $s2 , $s0  # Dia chi A[j+1] > A[i] ?
        beq     $t0, 0, dung    # if(false) thi ket thuc tra ve true
        lw      $t3,-4($s2)     # t3=A[i-1]
        lw      $t4,4($s2)      # t4=A[i+1]
        slt     $t6,$t3,$t4     # A[i]<A[i+1]           nguoc lai se la A[i+1]<=A[i-1]
        beq     $t6,$zero,sai   # neu t6!=0 thi tra ve false
        j       for             # neu khong thi quay lai vong for
sai:
```

```
        li      $v0, 59
        la      $a0, KetQua
        la      $a1, false
        syscall
        j       end_main
dung:
        li      $v0, 59
        la      $a0, KetQua
        la      $a1, true
        syscall
        j       end_main
err:
        sub     $s1, $s0, $s2   # s1 = A[i] - A[0]
        li      $t0, 4
        div     $s1, $t0
        mflo    $t4             # t4 = s1 / 4 : vi tri nhap sai
        add     $t4, $t4, 1

        li      $v0, 56
        la      $a0, Err
        add     $a1, $t4, $0
        syscall
        j       end_main
end_main:
```

Project 21.

- *Ý nghĩa các thanh ghi được sử dụng*

  $s1 = A
  $t0=count
  $a0 = string

- *Source code*

```
#Let's define the digit degree of some positive integer as the number of times we need to replace this number with the sum of its digits until we get to a one digit number.
#Given an integer, find its digit degree.
.data
Mess1:          .asciiz "Nhap so "
KetQua:         .asciiz "Ket Qua : "
Err:            .asciiz "Error "
Again:          .asciiz "Chay lai"
string:         .space 100

.text
main:
        li      $t0, 0          # count

        li      $v0, 54
        la      $a0, Mess1
        la      $a1, string
        la      $a2, 100
```

```
        syscall                 # Nhap so A

        la      $a0, string             # Lay phan tu string
        add     $t6, $0, $0             # j = 0

        jal     Check                   # kiem tra so nhap vao
        nop
        add     $s1, $0, $s0            # sumA = A
        jal     DigitDegree
        nop
the_end:
        li      $v0, 10
        syscall

Check:
        add     $t7, $a0, $t6           # $t7 = dia chi A lay ki tu dau tien cua string
        lb      $t2, 0($t7)             # $t2 = A
        li      $t7, 58                 # $t7 = '/'
        li      $a2, 47                 # $a2 = ':'
        beq     $t2, 10, end_Check      # if(A[j] == '\n') end check
        slt     $t3, $t2, $t7           # A[j] <= 9 ?
        slt     $t4, $a2, $t2           # A[j] >= 0 ?
        and     $t5, $t3, $t4           # A[j] <= 9 ? && A[j] >= 0 ?
        add     $t6, $t6, 1             # j++
        bne     $t5, 0, ADD1            # if(true) qua ADD1
        beq     $t5, 0, err             # else error

end_Check:
        li      $t0, 0          # sum = 0
        li      $t1, 10         # $t1=10
        jr      $ra
ADD1:
        sub     $t2, $t2, 1             # A[j] --
        sub     $t7, $t2, $a2   # $t7 = A[j] - 47
        mul     $s0, $s0, 10    # A = A*10
        add     $s0, $s0, $t7   # A += $t7
        j       Check

SumA:
        slt     $t2, $0, $s0    # 0 < A ?
        beq     $t2, $0, DigitDegree# if(A <= 0) nhay ve DigitDegree
        div     $s0, $t1                # A/10
        mfhi    $t3             # $t3 = A % 10
        mflo    $t4             # $t4 = A / 10
        add     $s1, $s1, $t3     # sumA += $t3
        add     $s0, $0, $t4    # A = $t4
        j       SumA
DigitDegree:
        add     $s0, $0, $s1    # A = sumA
        slt     $t2, $s0, $t1   # A < 10 ?
```

```
        bne     $t2, $0, end    # if(A < 10) end
        add     $t0, $t0, 1             # else : count += 1
        li      $s1, 0          # sumA = 0
        j       SumA
end:
        li      $v0, 56
        la      $a0, KetQua
        add     $a1, $t0, $0
        syscall
        jr      $ra

err:
        li      $v0, 59
        la      $a0, Err
        la      $a1, Again
        syscall
```

Project 24.

- *Ý nghĩa các thanh ghi được sử dụng*
  - $s6 : địa chỉ Array[i]
  - $s7 : địa chỉ Array[j]
  - $t0 : từ đứng trước
  - $t1 : từ đứng sau

- *Source code*

```
#Cyclone Word (challenge)
.data
Array:  .word
Mess1:.asciiz  "Nhap tu : "
Err:    .asciiz "Nhap sai loai "
Again: .asciiz  "Chay lai "
Mess2:          .asciiz "Cyclone Word"
Mess3:.asciiz "Not Cyclone Word"
KetQua:         .asciiz  ""
string:  .space  100
.text
main:
        jal Nhap
        nop

        jal Kiem_Tra
        nop

        j   end_main
#-----------------------------------------------
```

```
# Nhap : Nhap va kiem tra cac ky tu duoc nhap vao
#          va cho chung vao mot mang
# $s0 luu tru dia chi chu duoc nhap vao
# $s7 luu tru dia chi mang cac tu
#-------------------------------------------------
Nhap:
        li $v0, 54
        la $a0, Mess1
        la $a1, string
        la $a2, 100
        syscall                 # Nhap tu

        la $s0, string          # $s0 giu dia chi tu nhap vao
        la $s7, Array           # $s7 giu dia chi Array[0]
Check:
        lb  $t1, 0($s0)         # $t1 = ma Ascii cua string[i]
        beq $t1, 10, end_Check      # neu la ky tu ket thuc thi dung
        li  $t0, 96             # ky hieu truoc 'a'
        li  $t2, 123            # ky hieu sau 'z'
        slt $t3, $t0, $t1       # 'a' <= string[i] ?
        slt $t4, $t1, $t2       #  string[i] <= 'z' ?
        and $t5, $t3, $t4       # ('a' <= string[i] && string[i] <= 'z') ??
        beq $t5, 1, ADD             # if(true) thi them vao mang
        j  err                  # else bao loi
ADD:
        sw  $t1,0($s7)          # Array[i] = string[i]
        add $s7, $s7, 4             # i++
        add $s0, $s0, 1
        j   Check               # quay lai check
end_Check:
        sub $s7, $s7, 4
        jr  $ra
#-------------------------------------------------
# Kiem Tra : check xem tu nhap vao co phai Cyclone Word hay khong
#                 Bang cach kiem tra dan theo thu tu trai phai
# $s6 chay tu A[0] -> A[k]
# $s7 chay tu A[n] -> A[k]
#-------------------------------------------------
Kiem_Tra:
        la  $s6, Array
        lw  $t0, 0($s6)         # $t0 = A[0]
        lw  $t1, 0($s7)         # $t1 = A[i]
        li  $s2, 1              # $s2 = 1 ( tiep theo se xet phan tu ben trai)
        j   Compare             # so sanh $t0, $t1
Compare:
        slt $t2, $t1, $t0       # $t1 < $t0 ?
        beq $t2, 1, Not             # if($t1 < $t0) thi khong phai Cyclone Word (challenge)
        beq $s2, 1, Left        # $s2 = 1 (xet phan tu ben trai)
        j   Right               # $s2 = 0 (xet phan tu ben phai)
Left:
```

```
        add $t0, $t1, $0        # $t0 = $t1
        add $s6, $s6, 4                 # j++  chay tu trai qua phai
        beq $s6, $s7, end       # if(j = i) ket thuc
        lw  $t1, 0($s6)         # $t1 = Array[j]
        li  $s2, 0              # ( tiep theo se xet phan tu ben phai)
        j   Compare
Right:
        add $t0, $t1, $0        # $t0 = $t1
        sub $s7,$s7,4           # i --
        beq $s6, $s7, end       # if(i = j) ket thuc
        lw  $t1, 0($s7)         # $t1 = Array[i]
        li  $s2, 1              # ( tiep theo se xet phan tu ben trai)
        j Compare
end:
        li $v0, 59
        la $a0, KetQua
        la $a1, Mess2
        syscall
        jr $ra
Not:
        li $v0, 55
        la $a0, Mess3
        syscall
        jr $ra
err:
        li $v0, 59
        la $a0, Err
        la $a1, Again
        syscall
end_main:
```

KẾT QUẢ CHƯƠNG TRÌNH

Project 15.

- TH1: 1, a
  => Báo lỗi:

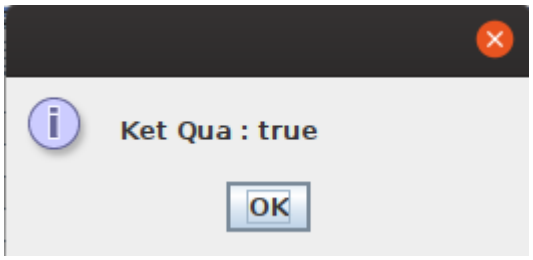
- TH2: 1,1,1

Project 16.

- TH1: 1, a
  => Báo lỗi:

- TH2: 1, 3, 2, 1

◆ TH3: 1,2,5,4

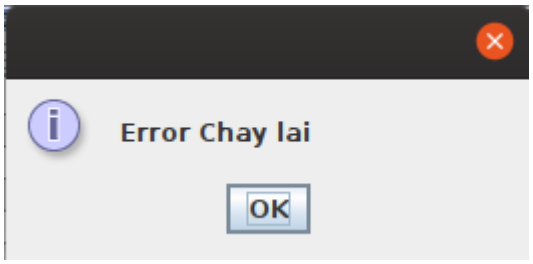


◆ TH4: 1,2,1,2

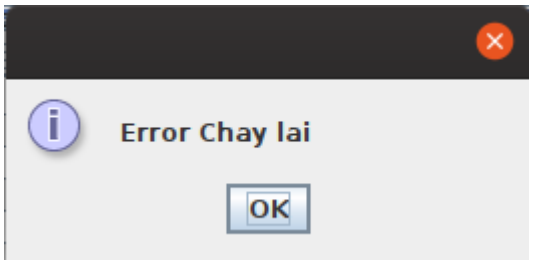
Project 21.

Nhập số:

◆ TH1: a

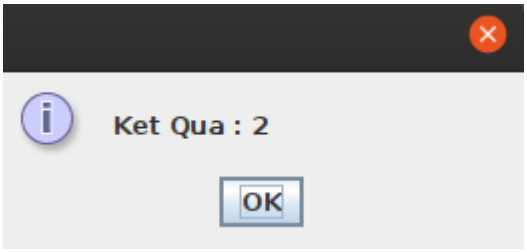


◆ TH2: -3



◆ TH3: 123

◆ TH4: 246



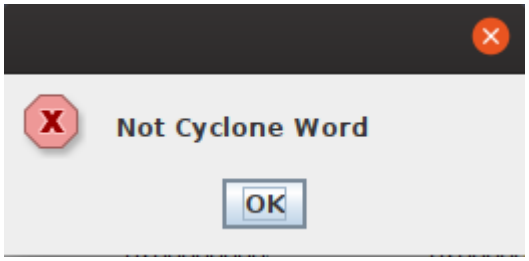
◆ TH5: 5

Project 24.

Nhập số

◆ TH1: 123

◆ TH2: abcds



◆ TH3: adjourned

◆ TH4: settled