

Báo cáo thực hành KTMT giữa kỳ

Họ và tên: Đỗ Gia Huy

MSSV: 20215060

Assignment 1 (Bài 20 chủ đề 1)

1. Yêu cầu đề bài

Nhập số nguyên N từ bàn phím, in ra màn hình số chính phương nhỏ nhất lớn hơn N .

2. Thuật toán thực hiện

B1: Nhập N

B2: Gán giá trị $index = 0$

B3: Cho số K cần tìm bằng $index * index$

B4: Nếu $K > N$ thì in kết quả K ra màn hình và kết thúc, ngược lại thì tăng giá trị $index$ lên 1 đơn vị và lặp lại bước 3

3. Code và giải thích các code

.data

mes1: .asciiz "Nhap N: "

mes2: .asciiz "Ket qua so chinh phuong lon nhat lon hon N: "

.text

li \$v0, 4 #In ra mes1

la \$a0, mes1

syscall

li \$v0, 5 #Nhap so N vao t1

syscall

move \$t1, \$v0

li \$s0, 0 #Gan gia tri index =0

loop:

```
    mul  $s1, $s0, $s0      #k = Index mu 2 la so can in ra
    slt  $t9, $t1, $s1      #Neu k>N thi $t9 = 1 , k<= N thi t9 =
0
    bne  $t9, $zero, endlp  #Neu $t9 = 1 khac 0 thi ket thuc vong
lap, duoc so k can tim
    addi $s0, $s0, 1        #Tang Index len 1 don vi
    j    loop
```

endlp:

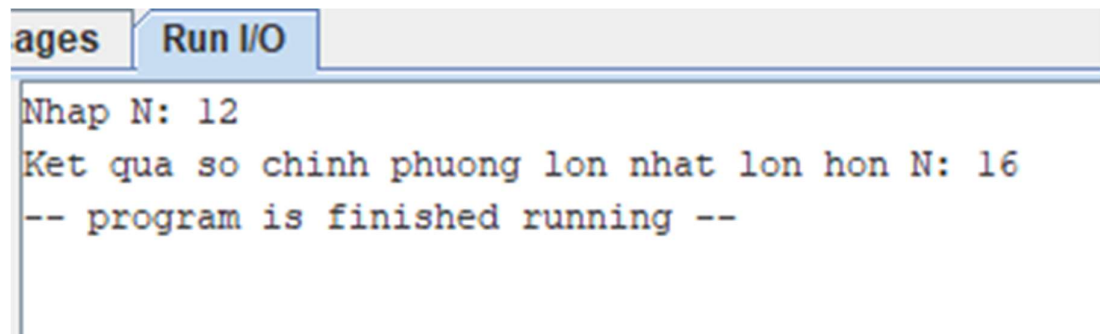
```
li    $v0, 4               #In mes 2
la    $a0, mes2
syscall
```

```
li    $v0, 1               #In so k
move  $a0, $s1
syscall
```

```
li    $v0, 10              #Ket thuc
syscall
```

4. Kết quả

- Với $N = 12 \Rightarrow K = 16 = 4*4$:



```
ages  Run I/O
Nhap N: 12
Ket qua so chinh phuong lon nhat lon hon N: 16
-- program is finished running --
```

- Với $N = 1424 \Rightarrow K = 1444 = 38*38$:

```
Nhap N: 1424
Ket qua so chinh phuong lon nhat lon hon N: 1444
-- program is finished running --
```

- Với $N = 6539 \Rightarrow K = 6561 = 81 \cdot 81$:

```
Nhap N: 6539
Ket qua so chinh phuong lon nhat lon hon N: 6561
-- program is finished running --
```

Assignment 2

1. Yêu cầu đề bài

Nhập mảng số nguyên từ bàn phím. In ra màn hình tổng các phần tử chẵn và tổng các phần tử lẻ trong mảng.

2. Mô tả thuật toán

B1: Nhập n là số phần tử mảng, nhập $A_1 \rightarrow A_n$ là các giá trị phần tử trong mảng

B2: Khai báo $t_{\text{chan}} = 0$ là tổng chẵn và $t_{\text{le}} = 0$ là tổng lẻ

B3: Duyệt lần lượt các giá trị $A_1 \rightarrow A_n$

- Nếu A_i chia 2 dư 0 thì $t_{\text{chan}} = t_{\text{chan}} + A_i$
- Nếu A_i chia 2 dư 1 thì $t_{\text{le}} = t_{\text{le}} + A_i$

Nếu $i = n$ thì duyệt xong A_n xong sẽ kết thúc bước 3

B4: In $t_{\text{chan}}, t_{\text{le}}$

3. Code và giải thích

3.1. Code:

.data

me1: .asciiz "Nhap n: "

me2: .asciiz "Khong phu hop! "

me3: .asciiz "Nhap cac phan tu: "

```
me4: .asciiz "Tong cac phan tu chan trong mang: "
```

```
me5: .asciiz "Tong cac phan tu le trong mang: "
```

```
nl: .asciiz "\n"
```

```
A: .word 0:100
```

```
.text
```

```
n_lp:
```

```
li    $v0, 4
```

```
la    $a0, me1
```

```
syscall
```

```
li    $v0, 5
```

```
syscall
```

```
move $t0, $v0
```

```
slti   $t1, $t0, 1                #Kiem tra loi, neu  $n < 1$  thi bi loi
```

```
bne    $t1, $zero, invalid
```

```
li     $t1, 0
```

```
la     $t2, A
```

```
la     $a0, me3
```

```
li     $v0, 4
```

```
syscall
```

```
li     $v0, 4
```

```
la     $a0, nl
```

```
syscall
```

```
j      input
```

invalid: #In loi va yeu cau nhap lai

la \$a0, me2

li \$v0, 4

syscall

la \$a0, nl

li \$v0, 4

syscall

j n_lp

input: #Nhap mang

beq \$t1,\$t0,inputted

li \$v0, 5

syscall

sw \$v0, 0(\$t2)

addi \$t1, \$t1, 1

addi \$t2, \$t2, 4

j input

inputted:

la \$t2, A

li \$t1, 0 # \$t1 la index

li \$t7, 2

li \$t8, 0 # \$t8 la tong gia tri chan

li \$t9, 0 # \$t9 la tng gia tri le

loop:

```

    beq  $t1, $t0, done
    lw   $s2, 0($t2)
    div  $s2, $t7    #Chia $s2 cho 2, phan nguyen nam o thanh $lo,
phan du nam o thanh $hi
    mfhi $s4          #Luu so du vao thanh $s4
    beq  $s4, $zero, chan
    add  $t9, $t9, $s2
    j    tiep
chan:
    add  $t8, $t8, $s2
    j    tiep
tiep:
    addi $t1, $t1, 1
    addi $t2, $t2, 4
    j    loop

```

done:

```

    li    $v0, 4
    la    $a0, nl
    syscall

```

```

    li    $v0, 4
    la    $a0, me4
    syscall

```

```

    li    $v0, 1
    move  $a0, $t8
    syscall

```

```
li    $v0, 4
```

```
la    $a0, nl
```

```
syscall
```

```
li    $v0, 4
```

```
la    $a0, me5
```

```
syscall
```

```
li    $v0, 1
```

```
move $a0, $t9
```

```
syscall
```

```
li    $v0, 10
```

```
syscall
```

3.2. Các chương trình con sử dụng:

n_lp: Chương trình con chuyên nhập n là số phần tử của mảng

invalid: Báo lỗi, yêu cầu nhập lại n

input: Nhập các phần tử trong mảng A

loop: Duyệt mảng A

chan: Cộng các số chẵn với trường hợp $A_i \bmod 2 = 0$

4. Kết quả thực thi

N = 14

A = 1 2 3 4 5 6 8 4 -6 9 2 -3 15 7

```
Nhap n: 14
Nhap cac phan tu:
1
2
3
4
5
6
8
4
-6
9
2
-3
15
7

Tong cac phan tu chan trong mang: 20
Tong cac phan tu le trong mang: 37
-- program is finished running --
```

Assignment 3

1. Yêu cầu đề bài

Nhập vào chuỗi ký tự. In ra các ký tự khác nhau có trong chuỗi.

2. Thuật toán

Duyệt từng ký tự một

3. Code

```
.data
```

```
string: .space 50
```

```
Message1: .ascii "Nhap chuoi: "
```

```
Message2: .ascii "So ki tu khac nhau trong chuoi la: "
```

```
Message3: .ascii "Cac ky tu khac nhau trong chuoi la: "
```

```
.text
```

```
main:
```


#Nhap chuoi

input:

li \$v0,54

la \$a0, Message1

la \$a1, string

li \$a2, 50

syscall

li \$s0,0 # count=0

la \$a0, string #a0 = address(string[0])

li \$t0,0 # i=0

jal check

nop

print:

add \$a1, \$zero, \$s0

li \$v0, 56

la \$a0, Message2

syscall

li \$v0, 4

la \$a0, Message3

syscall

li \$v0, 10 # exit

syscall

#Duyet tung ki tu cua chuoi

check:

```

add  $t2, $a0, $t0      # t2 = address(string[0] +i)
li    $s1,0             # appears = 0
lb    $s2,0($t2)         # s2 = string[i]
beq   $s2,$zero,end_check  # end when null
nop
li    $t1,0              # j=0

```

So sanh ki tu dang duyét voi cac ki tu truoc no trong chuoi

check1:

```

add  $t3, $a0, $t1      # t3 = address(string[0] +j)
lb    $s3, 0($t3)        # s3 = string[j]
bne   $s3,$s2,end_check1  # jump if s3 != s2 or string[j] !=
string[i]
nop
li    $s1,1             # appears =1
j     break
nop

```

end_check1:

```

addi  $t1,$t1,1    # j++
slt    $v0, $t1, $t0 # v0 =1 if j<i
bne    $v0, $zero, check1 # jump check1 when j<i
nop
j      break
nop

```

break:

```

addi  $t0, $t0, 1      # i++
bne   $s1,$zero,check  # when appears =0 => count++ else check
string[i++]
nop

```

```
addi $s0,$s0,1          # count++
```

```
li    $v0, 11
```

```
move $a0, $t2
```

```
syscall
```

```
j     check
```

```
nop
```

```
end_check:
```

```
jr    $ra
```

```
nop
```

4. Kết quả