





圖1

Input3

Output11

圖2

Intput15

Output1118

圖3

Intput21

Output59

4:

Intput4096

Output-1

5:

Intput-1

Output-1

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

文字說明:

第一段:

先設定好input和output 要cout 的string以及存放數值1,5,11,21,100在x5,x29,x6,x7,x30,以及把要output的最後結果值放在x18

Inputx:

用網站給的ECALL( la x10,inputx)將 intputx's address 讀到 x10 address

然後用li a7,4去印出input(也就是x10)的string("Input a number:")然後 li a7,5 去讀取

我們輸入的數字(在x10)然後用x19去存x10的int值然後去跑loop迴圈當loop跑完時用x10讀output的address來去印出outputx的string然後把跑完迴圈的x18的值給x10把它印出然後無條件離開到end去執行exit

F(x):

因為程式碼是由上到下讀取所以先把數值小的判別式先讀所以

x<0,x>99{先找<0的條件輸出都設為-1(18=-1),>99的條件輸出也設為-1 (18=-1)

然後當x19的數值為0或1時x18會分別加1和5回傳到loop}

{當x19的數值為1. 2~10, 2. 11~20 以及 3. 21~99

1.先借兩個stack空間去存x19和x1(分別是比較loop的值和回傳的值)然後去x19-1(f(x-1))去跑loop跑完後重新讀x19變成之前存的值以及還給stack4bit再做x19-2(f(x-2)) 去跑loop

跑完後還給stack4bit以及把x1變回之前的值jalr回去loop

2.和1.方法一樣差別只在x19減的數字是f(x-2)和f(x-3)

3. 先借兩個stack空間去存x19和x1然後把x19去做slli向左移動1bit相當於是\*2加到最後要輸出的x18然後重新讀x19去做除x29(5)(用 div x19,x19,x29)去跑loop迴圈跑完一樣重讀x1原本的值 jalr回去loop}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

.globl main

.data

endl: .string "\n"

inputx: .string "Input a number:"

outputx: .string "The damage:"

.text

main:

addi x18,x0,0 #計算總共傷害index

addi x5,x0,1 #x5=1

addi x6,x0,11 #x6=11

addi x7,x0,21 #x7=21

addi x29,x0,5 #x29=5

addi x30,x0,100#x30=100

#---------------------------------------------------------------------------------------inputx

la x10,inputx #Set x10 to intputx's address load x10 address(intputx)

li a7,4 #print the input(x10的intputx string)

ecall

li a7,5 #read the input(int)

ecall

add x19,x10,x0 #x19=inputx

jal x1,loop #呼叫到loop

la x10,outputx #讀取address

li a7,4 #print the input(x10的outputx string)

ecall

addi x10,x18,0 #將x18傷害的值給x10

li a7,1 #printint x10(a0)

ecall

beq x0,x0,end

end:

li a7,10 #exit

ecall

#---------------------------------------------------------------------------------------F(x)

loop:

blt x19,x0,funless #intputx

beq x19,x0,fun0 #intputx=0

beq x19,x5,fun1 #intputx=1

blt x19,x6,fun10 #intputx<11

blt x19,x7,fun20 #intputx<21

bge x19,x30,funover #intputx>=100

bge x19,x7,funbigger #intputx>=21

funless:

addi,x18,x18,-1

jalr x0,0(x1)

fun0:#intputx=0

addi x18,x18,1#damage+1

jalr x0,0(x1)

fun1:#intputx=1

addi x18,x18,5#damage+5

jalr x0,0(x1)

fun10:#intputx<11

addi x2,x2,-8 #stack先借兩個word的大小放intputx的值和回傳的值

sw x19,0(x2)

sw x1,4(x2)

addi x19,x19,-1#f(x-1)

jal x1,loop

lw x19,0(x2)#重新讀x19

addi x2,x2,4#stack上面word離開

addi x19,x19,-2#f(x-2)

jal x1,loop

lw x1,0(x2)#重新讀x1

addi x2,x2,4#stack上面word離開

jalr x0,0(x1)#回傳

fun20:#intputx<21

addi x2,x2,-8 #stack先借兩個word的大小放intputx的值和回傳的值

sw x19,0(x2)

sw x1,4(x2)

addi x19,x19,-2#f(x-2)

jal x1,loop

lw x19,0(x2)#重新讀x19

addi x2,x2,4#stack上面word離開

addi x19,x19,-3#f(x-3)

jal x1,loop

lw x1,0(x2)#重新讀x1

addi x2,x2,4#stack上面word離開

jalr x0,0(x1)

funbigger:#intputx>=21

addi x2,x2 -8 #stack先借兩個word的大小放intputx的值和回傳的值

sw x19,0(x2)

sw x1,4(x2)

slli x19,x19,1#x乘2

add x18,x18,x19 #傷害=原本傷害+x\*2

lw x19,0(x2)#重新讀x19

div x19,x19,x29#x除5

jal x1,loop

lw x1,4(x2)#重新讀x1

addi x2,x2,8#stack上面兩個word離開

jalr x0,0(x1)

funover:

addi x18,x18,-1

jalr x0,0(x1)