

计算机组成原理实验报告三

姓名： 卫焱滨

学号： 11710823

一、实验内容

1. 将 400 以内满足 $t = x*x + y*y + z*z + xy + xz + yz$ (x, y, z 为正整数) 的 t 值打印输出.
2. 将一个字符串 (除字符串结束符外的其他字符) 倒序存储,再打印输出

二、实验步骤 (阐述代码思路或操作步骤)

1.

题意分析:

本题要求输出 400 以内的满足 $t = x*x + y*y + z*z + xy + xz + yz$ (x, y, z 为正整数) 的 t 值

操作思路和步骤 :

1. 分析 xyz 取值范围: 由于 $1 \leq t \leq 400$, 且 $t \geq x^2, t \geq y^2, t \geq z^2$, 因此可以将 x, y, z 的范围限定在 1 到 20 之间
2. 循环枚举 t : 构建三层嵌套循环, 最外层从 1 到 20 枚举 x , 次外层从 x 到 20 枚举 y , 最内层从 y 到 20 枚举 z
3. 判断合法 t : 在最内层循环里, 对于每种 x, y, z , 求得相应的 t , 并判断其与 400 的相对大小, 确定其是否合法, 由于 1 中的分析, 此时可保证所有合法的 t 不会遗漏
4. 记录合法 t : 在内存中创建一个大小为 410, 初始值为 0 的 byte 数组, 有效部分下标为 0~399, 分别代表下标代表的数是否为合法的 t . 对于 4 中的每个合法的 t , 计算得到数组相应下标所对应的地址位置, 将该位置的数组值修改为 1
5. 打印输出所有合法 t : 此时数组中记录了所有合法的 t 的信息, 遍历数组, 若值为 1, 输出其下标和空格, 继续循环, 若值为 0, 不输出直接继续循环, 直到数组遍历结束。

代码 :

.data :

```
arr: .space 410
space: .asciiz " "
en: .asciiz "\n"
```

.text :

```
main:
la $s0 arr
li $t0 1
li $t1 1
li $t2 1
```

LPX: addi \$t1 \$t0 0

#loop for x from 1 to 20

LPY: addi \$t2 \$t1 0

#loop for y from x to 20

LPZ: li \$t3 0

#loop for z from y to 20

li \$t4 0

mul \$t3 \$t0 \$t0

#compute $t = x*x + y*y + z*z + xy + xz + yz$

add \$t4 \$t4 \$t3

mul \$t3 \$t1 \$t1



```
        add $t4 $t4 $t3
        mul $t3 $t2 $t2
        add $t4 $t4 $t3
        mul $t3 $t0 $t1
        add $t4 $t4 $t3
        mul $t3 $t0 $t2
        add $t4 $t4 $t3
        mul $t3 $t1 $t2
        add $t4 $t4 $t3
        li $t6 400                                #identify whether t satisfy
        slt $t3 $t4 $t6
        bne $t3 1 c
        add $t5 $s0 $t4
        li $t3 1
        sb $t3 ($t5)
        c:addi $t2 $t2 1
        ble $t2 20 LPZ                            #control z loop
    addi $t1 $t1 1
    ble $t1 20 LPY                                #control y loop
addi $t0 $t0 1
ble $t0 20 LPX                                  #control x loop

#Traverse to print
li $s1 0
li $s5 1
wrp:    add $s2 $s1 $s0    #loop to traverse array to print t
        lb $s3 ($s2)
        bne $s3 1 n
        li $v0 1
        addi $a0 $s1 0
        syscall
        addi $s5 $s5 1
        beq $s5 20 L      #if print 20 numbers, then change a line
        la $a0 space
        li $v0 4
        syscall
        j n
L:li $s5 0                #change a line
    la $a0 en
    li $v0 4
    syscall
    j n
```

```
n:addi $s1 $s1 1      #control loop
ble $s1 400 wrp
```

2.

题意分析:

将一个字符串（除字符串结束符外的其他字符）倒序存储,再打印输出

操作步骤与思路：

1. 调用 syscall，让用户输入字符串，所输入的字符串存储于内存中的数组 str 中
2. 循环寻找'\0'：遍历数组直到找到结束符'\0'，记录结束符'\0'在内存中出现的地址和 str 的首地址
3. 循环交换：用两个寄存器\$a0, \$s1，分别代表左指针，和右指针，两者的初值分别是内存中字符串数组 str 的首地址，和最后一个字符的地址（2 中记录的结束符地址-1 得到），利用 lb 和 sb 交替操作交换两者所指向位置的字符，每次循环将左指针\$a0 地址+1，右指针 \$s1 地址-1，当左右指针相遇时退出循环
4. 打印输出：完成交换后，调用 syscall 打印输出新生成的倒序字符串

代码：

```
.data
    str: .space 100
.text
    li $a1 99
    li $v0 8
    la $a0 str
    li $s0 0
    syscall

#find the '\0' s address
count:add $s1 $s0 $a0
    lb $s2 ($s1)
    beq $s2 0 Inv
    addi $s0 $s0 1
    j count
    # $s1 is address of \0
```

#using two pointer registers to swap

```
Inv:    subi $s1 $s1 1
    lb $t0 ($a0)
    lb $t1 ($s1)
    sb $t0 ($s1)
    sb $t1 ($a0)
    addi $a0 $a0 1
    blt $a0 $s1 Inv
```

```
#print the new string
```

```
print:li $v0 4
```

```
la $a0 str
```

```
syscall
```

```
exit: li $v0 10
```

```
syscall
```

三、实验结果（截图并配以适当的文字说明）

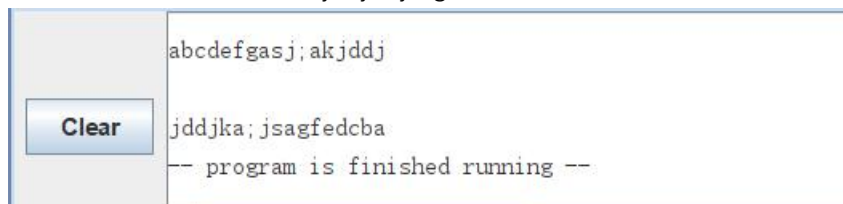
1.

运行结果如下，共 249 个满足题目要求的 t 数（不包含 400 在内）

```
6 11 17 18 24 25 27 33 34 35 38 43 44 45 47 51 54 55 57
58 61 66 67 68 69 71 72 73 77 81 82 83 85 86 87 89 90 95 96
97 99 100 102 103 105 108 109 113 114 115 117 118 119 121 122 123 125 129 130
131 132 133 135 136 137 139 140 141 143 145 146 147 150 151 152 153 154 157 159
161 162 163 166 169 171 172 173 175 176 177 178 179 180 181 183 185 187 188 191
193 194 195 197 198 199 201 202 203 204 205 207 209 210 213 214 215 216 217 218
219 220 223 225 227 228 229 232 233 234 235 237 241 242 243 244 245 246 247 249
250 251 253 255 257 258 261 262 263 264 265 266 267 268 269 271 272 273 274 275
276 277 278 279 281 283 284 285 287 288 289 291 292 293 294 295 297 298 299 301
303 306 307 308 310 311 313 315 319 321 322 323 324 325 326 327 328 329 330 331
332 333 335 337 339 340 342 343 344 345 346 347 348 349 353 354 355 356 357 358
359 360 361 362 363 365 367 369 370 373 374 375 377 378 379 380 383 384 385 386
387 388 389 390 391 393 394 395 396 397
-- program is finished running (dropped off bottom) --
```

2.

举例测试运行结果如下，若测试输入字符串 abcdefgasj;akjddj，则会以相反字符顺序输出 jddjka;jsagfedcba



```
abcdefgasj;akjddj
jddjka;jsagfedcba
-- program is finished running --
```

四、实验分析（遇到的问题以及解决方案）

问题 1: Lab3_1 运行时间过长，需要运行将近 50s 才会出现结果

解决: 优化循环结构，改变循环方式

将 x, y, z 都为 1~20 的循环，优化为 x 从 1~20, y 从 x~20, z 从 y~20 的循环，可在保证正确性的同时极大减少运行时间

问题 2: Lab3_1 输出结果过长, 屏幕无法装下

解决: 引入 counter, 运用分支结构, 在输出时进行判断。每 20 个数字进行换行操作