CS

Started: 2019.11.12 11:23:49

Submitted: 2019.11.21 13:05:06

Score: 100

DigiPen DigiPen

CS100A

Assignment #8



Demonstrate the ability to write assembly code with flow control.

Vadim Surov

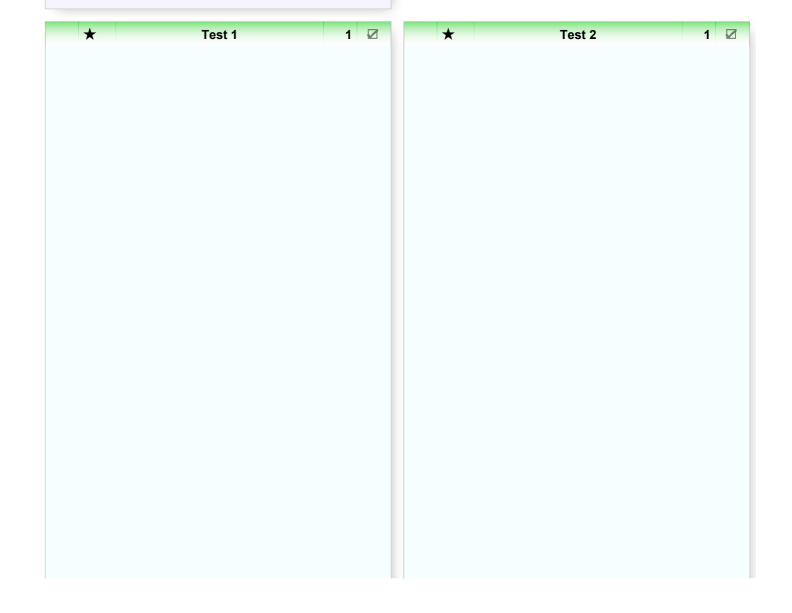
Rules

Read carefully and check all rules you agree with:

- Each card has description which must be strictly followed.
- Keep the code properly formatted (correct indentation, line width is 40 characters max).
- Your code must represent your own individual work. If something is not clear, ask your instructor for help.
- Cheating of any kind (copying someone else's work, allowing others to copy your work, collaborating, etc.) will not be tolerated and will be dealt with SEVERELY.

Problem

- Given two circles with integral centers (x1,y1), (x2,y2), and radii r1, r2.
- Create the code that determins intersection or nonintersection of the circles.
- You code must output "intersecting" or "nonintersecting" based on the result of calculation.
- Place your the same code in the middle of the following test cases. Do not modify given values in the tests.
- Use the most efficient method (without square root calculations)



```
Run
   .data
x1:
       .long -5
      -6
y1:
       .long 20
r1:
x2:
      .long 7
      .long 8
y2:
             30
      .long
strl: .asciz "intersecting"
str2: .asciz "non-intersecting"
    .text
    .global main
main:
   push %rbx # For alignment
   mov x2, %eax
   sub x1, %eax
   imul %eax, %eax
   mov y2, %ecx
   sub y1, %ecx
   imul %ecx, %ecx
   add %eax, %ecx
   mov r1, %edx
   add r2, %edx
   imul %edx, %edx
   CMP %ecx, %edx
    JG nonintersect
   mov $str2, %rdi
    call printf
   jmp end
nonintersect:
   mov $str1, %rdi
   call printf
  end:
          %eax, %eax # return 0;
   xor
   pop
          %rbx
   ret
```

```
v2:
       .long 40
             4
r2:
       .long
str1: .asciz "intersecting"
str2: .asciz "non-intersecting"
    .text
    .global main
main:
    push %rbx # For alignment
    mov x2, %eax
    sub x1, %eax
    imul %eax, %eax
    mov y2, %ecx
    sub y1, %ecx
    imul %ecx, %ecx
    add %eax, %ecx
    mov r1, %edx
    add r2, %edx
    imul %edx, %edx
    CMP %ecx, %edx
    JG nonintersect
    mov $str2, %rdi
    call printf
    jmp end
nonintersect:
   mov $str1, %rdi
    call printf
  end:
          %eax, %eax # return 0;
    xor
    pop
           %rbx
    ret
non-intersecting
```

Run

x1:

y1:

r1:

x2:

.data

-10

.long -20

.long 2

.long 30

intersecting

Survey

• What is approximate number of hours you spent implementing this assignment?

1hr

• Indicate the specific portions of the assignment that gave you the most trouble

Test 2

By signing this document you fully agree that all information provided therein is complete and true in all respects.

Responder sign:

Copyright © 2019 | Powered by MyTA | www.mytaonline.com