; Add20Data.asm

.include "m328Pdef.inc"

init:

.def i = r17

.def summing = r18

.def sumLow = r20

.def sumHigh = r21

.def outB = r23

.def outC = r24

.def zero = r25

main:

clr i

ldi r30, low(0x100)

ldi r31, high(0x100)

clr sumLow

clr sumHigh

clr summing

ser r23

ser r24

out DDRB, r23

out DDRC, r24

clr zero

loop:

cpi i, 20

brge end

ld summing, Z+

add sumLow, summing

adc sumHigh, zero

inc i

rjmp loop

end:

out PortB, sumLow

out PortC, sumHigh

ret

; CompTwo.asm

.include "m328Pdef.inc"

.def first = r17

.def second = r18

main:

ldi r30, low(0x100)

ldi r31, high(0x100)

ser r16

out DDRB, r16

ld first, Z+

ld second, Z+

cp first, second

brge firstIsGreater

rjmp secondIsGreater

firstIsGreater:

out PortB, first

rjmp end

secondIsGreater:

out PortB, second

end:

ret

; Output10.asm

.include "m328Pdef.inc"

.def a = r17

.def i = r18

init:

clr i

ldi zl, low(0x100)

ldi zh, high(0x100)

ser r16

out DDRB, r16

loop:

cpi i, 10

brge end

ld a, Z+

out PortB, a

inc i

rjmp loop

end:

ret

; Sum20.asm

.include "m328Pdef.inc"

init:

.def i = r16

.def n = r17

.def sum = r18

main:

ldi i, 1

ldi n, 0x14 ;load 20 into n

clr sum

loop:

cp n, i

brlo end

add sum, i

inc i

rjmp loop

end:

out PortB, sum

ret

; Add1ToN.asm

.include "m328Pdef.inc"

init:

.def n = r16

.def i = r17

.def sumLow = r20

.def sumHigh = r21

.def outB = r23

.def outC = r24

.def zero = r25

main:

ldi i, 0x01

ldi zl, low(0x100)

ldi zh, high(0x100)

ld n, Z+

clr sumLow

clr sumHigh

ser r23

ser r24

out DDRB, r23

out DDRC, r24

clr zero

inc n

loop:

cp i, n

brge end

add sumLow, i

adc sumHigh, zero

inc i

rjmp loop

end:

out PortB, sumLow

out PortC, sumHigh

ret

; Comp20.asm

.include "m328Pdef.inc"

init:

.def i = r17

.def highest = r18

.def checking = r19

.def outB = r23

.def outC = r24

main:

ldi i, 0x01

ldi r30, low(0x100)

ldi r31, high(0x100)

clr highest

clr checking

ser r23

out DDRB, r23

ld highest, Z+

loop:

cpi i, 20

brge end

inc i

ld checking, Z

cp checking, highest

brge overwriteHighest

ld checking, Z+

rjmp loop

overwriteHighest:

ld highest, Z+

rjmp loop

end:

out PortB, highest

ret

; InClassQuiz.asm

init:

.def a = r16

.def b = r17

.def result1 = r18

.def result2 = r19

.def productResult = r20

.def doubleResult = r21

main:

ldi r28, low(RAMEND-3)

ldi r29, high(RAMEND-3)

out SPH, r29

out SPL, r28

ldi a, 0x06

std Y+1, a

ldi b, 0x04

std Y+2, b

clr productResult

call product

ldd productResult, Y+3

clr doubleResult

std Y+3, productResult

call doublenum

ldd doubleResult, Y+4

out PortB, doubleResult

rjmp end

product:

ldd a, Y+1

ldd b, Y+2

mul a, b

std Y+3, r0

ret

doublenum:

ldd productResult, Y+3

lsl productResult

std Y+4, productResult

ret

end:

ret