

;código referente a atividade número 2

loop:

Sx->Ac+	n	;carrega o n
At->S(x)	x	;guarda x com n
S(x)->Ac+	n	;carrega o n
Cc->S(x)	pos	;se o Ac>=0, pula para pos
halt		
.empty		

verif1:

S(x)->Ac+	vet	;carrega o vetor
A/S(x)->R	x	;divide por x
S(x)->Ah-	const1	;subtrai o resultado da divisão por 1
Cc->S(x)	verif3	;Se AC>=0, salta para a label verif3
Cu->S(x)	verif2	;SENAO, pula para a label verif2

verif2:

At->S(x)	cont	;carrega o contador
S(x)->Ah-	one	;subtrai 1
Cc->S(x)	verif3	;se AC>=0, volta para a label verif3
S(x)->Ac+	const2	;SENAO, carrega o 2 no contador novamente
At->S(x)	cont	;o numero não é um primo
Cu->S(x)	loop	;volta para o inicio

verif3:

S(x)->Ac+	x	;carrega o x para subtrai-lo
S(x)->Ah-	const1	;atualiza o x
At->S(x)	x	;guarda o novo valor
S(x)->Ac+	x	;carrega x
S(x)->Ah-	const1	;subtrai por 1 para testar se o resultado
Cc->S(x)	verif	;se AC>=0, pula para verif1
S(x)->Ac+	cont	;carrega cont para verificar se há
		;apenas 2 numeros com resto 0
S(x)->Ah-	const1	;subtrai por 1 para verificar se o cont
		;está com o valor 0
Cc->S(x)	loop	;se AC>=0, pula para o inicio
S(x)->Ac+	prim	;SENAO, carrega a variavel que guarda o
		;num de primos
S(x)->Ah+	const1	;adiciona 1 num primo
At->S(x)	prim	;atualiza o num de primos
S(x)->Ac+	const2	;carrega o valor 2
At->S(x)	cont	;atualiza o cont com 2 de novo
Cu->S(x)	loop	;volta para o inicio

pos:

At->S(x)	vet	;guarda n no vetor
S(x)->Ac+	n	;carrega n
S(x)->Ah-	const1	;decrementa 1
At->S(x)	n	;atualiza o tamanho de n
Cu->S(x)	verif	;pula para a label verif1

x:	.data 10	;auxiliar
n:	.data 10	;vetor de tamanho 10
vet:	.data 100	;vetor
const1:	.data 1	;constante 1
const2:	.data 2	;constante 2
cont:	.data 2	;contador
prim:	.data 12C	;armazena o numero de primos