E-fólio B | Folha de resolução para E-fólio

UNIDADE CURRICULAR: Arquitetura de Computadores

CÓDIGO: 21010

DOCENTE: José Coelho

A preencher pelo estudante

NOME: Diogo Miguel Palma Sustelo

N.º DE ESTUDANTE: 2201148

CURSO: Licenciatura em Engenharia Informática

TRABALHO / RESOLUÇÃO:

Explicação alinea A:

Na alínea A, começo por inicializar a variável R6 a 0, para ser comparada com a variável R1 que contem o numero de primos pretendido, e parar quando chegar a esse número, indo sendo incrementada à medida que um numero primo é encontradao, e R2 inicializado a 1 e incrementado em mais 1, por forma a quando faz a validação, se for menor ou igual a 3, é considerado primo e uma vez que foi inicializada a variável a 1, o 1 é desconsiderado.

Na função “VerificaPrimo” além da verificação de ser menor ou igual a 3, verifica também se é divisível por eles, umas vez que se for divisivel, não é primo e descarta esses números.

Caso não sejam divisíveis por 2 ou 3, é depois verificado na função VerificaPrimo2, através de um loop, se é divisível por os restantes números primos, em caso sendo, são também desconsiderados.

Caso seja numero primo é colocado em memória com o auxilio da função EPrimo.

Explicação alinea B:

Na alinea B começa-se por alojar os valores de K e W fornecidos, indo buscar a dois segmentos de memória seguidos e colocando nas variáveis R3 e R4 e fazendo uma cópia para R6 e R7, uma vez que depois com as divisões e subtrações os valores das variáveis são destruídos.

Na função VerificaFator verifica se K é 0 ou 1 e termina a função, caso não seja, faz a divisão de K por W, e caso seja fator, passa para a função VericaFPrimo, para verificar se também é primo, com auxilio da lista criada na alinea anterior.

Caso K/W não seja fator, o programa prossegue e vai fazer a divisão de K/(K-W), e novamente se for fator, irá verificar também se é primo.

Caso nenhuma das opções seja verdade, então a jogada não é válida, caso seja válida, é incrementado valor à variável R1, de modo a no final passar o valor do numero de jogadas.

Explicação alinea C:

Nesta alinea teríamos de verificar quais números primos se aplicariam e seriam então divisores do numero fornecido, e com esses números construir jogadas com K/W e K-W, a função JogadasJogoPrimo recebe o valor de R1 que contém o valor de K a ser considerado para a jogada, após o qual é necessário criar as jogadas.

Para tal foi feita uma função CorrePrimos, que percorre a lista dos números primos até ao valor de K, e se verifica quais são divisores de K, através da verificação da divisão inteira, caso seja divisor de K, é porque é fator primo e por isso cria-se as jogadas passando para a função CFatorPrimo, onde se coloca sequencialmente em memória os valores obtidos da divisão e da subtração de K e W.

Explicação alinea D:

Nesta alinea teríamos de verificar qual a melhor jogada a partir do numero K, para isso coloquei o numero K numa chamada à alinea anterior, e após isso com a função JogadaMemoria, coloquei todos os valores possíveis de jogada na pilha, os quais fui testando sequencialmente também na chamada à alinea C novamente e às quais a fui retirando e comparando sequencialmente na função ComparaJogada, e caso fosse maior o numero de jogadas ou igual mas o numero fosse maior, assumia esse valor na função SubstituiJogada, que no final do programa ia ser passado para R1 para retornar ao programa principal.

Anexos:

Código do programa:

```
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            ; Início do bloco A vvvvvvvv não alterar para baixo vvvvvvvv
            ; zona de dados
            ORIG 8000h
resultadosA     TAB     16
resultadosB     TAB     16
resultadosC     TAB     16
resultadosD     TAB     16
            ; jogos de teste para a alínea B
Jogo1           STR     476,68,66,63,9,6,3,1,0
Jogo2           STR     476,469,156,78,39,13,3,0
Jogo3           STR     1547,119,102,83,27,1
Jogo4           STR     4389,4370,2185,87,80,40,5,1
Jogo5           STR     3990,570,30,10,2,0
Jogo6           STR     833,17,0
Jogo7           STR     798,399,392,196,49,7,1
Jogo8           STR     1254,114,112,110,108,54,27,9,6,4,2,1
Jogo9           STR     884,68,34,32,16,14,7,1
Jogo10          STR     180,60,56,28,4,2,1
Jogo11          STR     1001,143,132,66,6,3,0
Jogo12          STR     931,912,910,455,91,1
Jogo13          STR     34,17,1
Jogo14          STR     1862,1813,906,453,1
Jogo15          STR     1020,510,255,51,1
Jogo16          STR     3762,3759,537,31,1
            ; colocar jogos
Jogos           STR Jogo1,Jogo2,Jogo3,Jogo4,Jogo5,Jogo6,Jogo7,Jogo8,Jogo9,Jogo10,Jogo11,Jogo12,Jogo13,Jogo14,Jogo15,Jogo16,0
            ; números de teste para a alínea C e D
Numeros         STR 476,66,5179,34,77,24,40,12,1155,6175,2520,1716,1377,7225,7,5,0
            ; zona para retornar as jogadas
jogadas         TAB     16
            ; zona para colocar os números primos
            ; os jogos não podem ter números superiores ao maior primo
maxNumeros      EQU     1024
primos          TAB     maxNumeros
            ; Final do bloco A ^^^^^^^^^ não alterar para cima ^^^^^^^^^
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;





            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            ; Início do bloco B vvvvvvvv não alterar para baixo vvvvvvvv
            ; zona do código
            ORIG 0000h

            ; inicialização do stack
Inicio:                 MOV R1, fd1fh
            MOV SP, R1


            ; TESTE alinea A
            MOV R1, maxNumeros
            CALL PrimeirosPrimos
            MOV R2, resultadosA
            MOV R4, 16
CicloA:         DEC R1
            MOV R5, M[R1+primos]
            MOV M[R2], R5
            INC R2
            DEC R4
            BR.NZ CicloA
FimA:           Nop

            ; TESTE alinea B
            MOV R1, Jogos
            MOV R4, R0
CicloB:         MOV R2, M[R1]
            CMP R2, R0
            BR.Z FimB
            PUSH R1
            PUSH R4
            CALL ValidarJogoPrimo
            POP R4
            MOV M[R4+resultadosB], R1
            POP R1
            INC R4
            INC R1
            BR CicloB
FimB:           Nop


            ; TESTE alinea C
            MOV R5, Numeros
            MOV R4, R0
CicloC:         MOV R2, jogadas
            MOV R1, M[R5]
            CMP R1, R0
            BR.Z FimC
```

```
            PUSH R4
            PUSH R5
            CALL JogadasJogoPrimo
            POP R5
            POP R4
            MOV M[R4+resultadosC], R0
CicloC2:        DEC R1
            BR.N CicloC3
            MOV R3, M[R1+jogadas]
            ADD M[R4+resultadosC], R3
            BR CicloC2
CicloC3:        INC R5
            INC R4
            BR CicloC
FimC:           Nop

            ; TESTE alinea D
            MOV R5, Numeros
            MOV R4, R0
CicloD:         MOV R1, M[R5]
            CMP R1, R0
            BR.Z FimD
            PUSH R4
            PUSH R5
            CALL JogoArtificialPrimo
            POP R5
            POP R4
            MOV M[R4+resultadosD], R1
            INC R5
            INC R4
            BR CicloD
FimD:           JMP Fim
            ; Final do bloco B ^^^^^^^^^ não alterar para cima ^^^^^^^^^
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            ; função PrimeirosPrimos solicitada na alínea A
            ; Entrada: R1 - número de primos pretendido;
            ;          primos - endereço do local onde guardar os números
            ; Saída: resultado no endereço em R2
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
PrimeirosPrimos:MOV R6,R0
            MOV R2,1
ContaPrimos:    INC R2
            CMP R6,R1
            BR.N VerificaPrimo
            RET

VerificaPrimo:  MOV R3,3
            CMP R2,R3
            BR.NP EPrimo
            MOV R3,R2
            MOV R4,2
            DIV R3,R4
            CMP R4,R0
            BR.Z ContaPrimos
            MOV R3,R2
            MOV R4,3
            DIV R3,R4
            CMP R4,R0
            BR.Z ContaPrimos
            MOV R3,5
            BR LoopPrimo

EPrimo:         MOV M[R6+primos], R2
            INC R6
            BR ContaPrimos

VerificaPrimo2: MOV R4,R2
            MOV R5,R3
            DIV R4,R5
            CMP R5,R0
            BR.Z ContaPrimos
            MOV R4,R2
            MOV R5,R3
            ADD R5,2
            DIV R4,R5
            CMP R5,R0
            JMP.Z ContaPrimos
            ADD R3,6
            BR LoopPrimo

LoopPrimo:      MOV R4,R3
            MOV R5,R3
            MUL R4,R5
            CMP R2,R5
            BR.NN VerificaPrimo2
            BR EPrimo

            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            ; função ValidarJogoPrimo solicitada na alínea B
            ; Entrada: R2 - endereço com as jogadas
            ;          primos - endereço com os primeiros números primos
            ; Saída: R1 - número de jogadas válidas
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ValidarJogoPrimo:MOV R1, R0
AlojaKW:        MOV R3, M[R2]
            INC R2
            MOV R4, M[R2]
```

```
            MOV R6,R3
            MOV R7,R4


VerificaFator:  CMP R3,R0
            BR.Z Retorna
            MOV R5,1
            CMP R3,R5
            BR.Z Retorna

            DIV R3,R4
            CMP R4,R0
            BR.Z VerificaFPrimo

            MOV R3,R6
            MOV R4,R6
            MOV R5,R7
            SUB R3,R5
            MOV R5,R3
            DIV R4,R5
            CMP R5,R0
            BR.Z VerificaFPrimo
            BR Retorna

VerificaFPrimo: MOV R4,R0
LoopVFPrimo:    CMP R4,maxNumeros
            BR.Z Retorna
            MOV R5,M[R4+primos]
            CMP R3,R5
            BR.N Retorna
            BR.Z EFatorPrimo
            INC R4
            BR LoopVFPrimo

EFatorPrimo:    INC R1
            JMP AlojaKW

Retorna:        RET


            ..............................................
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            ; função JogadasJogoPrimo solicitada na alínea C
            ; Entrada: R1 - número K a processar
            ;         R2 - endereço no qual as jogadas devem ser colocadas
            ;         primos - endereço com os primeiros números primos
            ; Saída: R1 - número de jogadas; resultado no endereço R2
            ..............................................
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
JogadasJogoPrimo:MOV R3,R0
            MOV R4,R1
            MOV R1,R0

CorrePrimos:    CMP R3, maxNumeros
            BR.Z RetornaC
            MOV R5,R4
            MOV R6,M[R3+primos]
            MOV R7,M[R3+primos]
            CMP R5,R6
            BR.NP RetornaC
            DIV R5,R6
            INC R3
            CMP R6,R0
            BR.Z CFatorPrimo
            BR CorrePrimos

CFatorPrimo:    MOV M[R2],R5
            INC R2
            INC R1
            MOV R5,R4
            SUB R5,R7
            CMP R5,1
            BR.Z CorrePrimos
            MOV M[R2],R5
            INC R2
            INC R1
            BR CorrePrimos

RetornaC:       RET


            ..............................................
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            ; função JogoArtificialPrimo solicitada na alínea D
            ; Entrada: R1 - número K a processar
            ;         primos - endereço com os primeiros números primos
            ; Saída: R1 - valor W, com a jogada para K
            ..............................................
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
JogoArtificialPrimo:MOV R2, jogadas
            CALL JogadasJogoPrimo
            MOV R3,R0
            MOV R4,R0
            MOV R5,R1
            MOV R6,R0
            MOV R7,R0

JogadaMemoria:  CMP R7,R1
            BR.NN ComparaJogada
            PUSH M[R7+jogadas]
            INC R7
            BR JogadaMemoria


ComparaJogada:  CMP R6,R5
```

```
                BR.Z RetornaD
                POP R1
                PUSH R1
                PUSH R3
                PUSH R4
                PUSH R5
                PUSH R6
                MOV R2, jogadas
                CALL JogadasJogoPrimo
                POP R6
                POP R5
                POP R4
                POP R3
                INC R6
                CMP R3,R1
                BR.NP SubstituiJogada
                POP R1
                BR ComparaJogada

SubstituiJogada:MOV R3,R1
                POP R1
                CMP R4,R1
                BR.NN ComparaJogada
                MOV R4,R1
                BR ComparaJogada

RetornaD:       MOV R1,R4
                RET


                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ; Início do bloco C vvvvvvvv não alterar para baixo vvvvvvvv
                ; última instrução, não alterar
Fim:               JMP Fim
                ; Final do bloco C ^^^^^^^^^ não alterar para cima ^^^^^^^^^
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Conteúdo da memória:

```
8000 : 1fe1 1fd3 1fbb 1fb5 1faf 1fa5 1f9d 1f99 ........
8008 : 1f97 1f91 1f85 1f7b 1f75 1f67 1f51 1f4b ........
8010 : 0007 0001 0002 0002 0005 0000 0003 000b ........
8018 : 0007 0001 0006 0004 0002 0000 0003 0002 ........
8020 : 06c8 00f3 0000 0044 009a 003f 0065 001d ...D.?e.
8028 : 1568 502b 32e3 2169 0cca 3faa 0000 0000 ........
8030 : 01da 003f 0000 0020 0046 0016 0026 000a .?..F.&.
8038 : 047e 1812 09d1 06a9 055e 1c34 0000 0000 ........
```

Número de instruções e ciclos de relógio:

Alinea A

Número de instruções: 366907

Ciclos de relógio: 9284251

Alinea B

Número de instruções: 3214

Ciclos de relógio: 30988

Alinea C

Número de instruções: 44731

Ciclos de relógio: 527648

Alinea D

Número de instruções: 172955

Ciclos de relógio: 2037586

Total do programa

Número de instruções: 587807

Ciclos de relógio: 11880473