

```

1 ;Arquitetura de Computadores
2 ;Projeto 2
3 ;Alunas:
4 ;Érica Filipa Dias Cunha nº 2033617
5 ;Jocelyne Estefanía Pestana nº 2036617
6
7 ;Periféricos
8 ON_OFF EQU 1A0H ;botão on/off
9 SEL_NR_MENU EQU 1B0H ;botão de seleção
10 OK EQU 1C0H ;botão ok
11 CHANGE EQU 1D0H ;botão change
12 PESO EQU 1E0H ;botão para inserir o peso do
    alimento que queremos
13
14 ;Posição do Display
15 Display_inic EQU 200H ;Inicio do display principal
16 Display_end EQU 26FH ;fim do display principal
17
18 ;Posições onde o alimento vai ser escrito no menu principal da balança
19 Linha_inic EQU 0220H ;Inicio da linha onde começa o
    alimento a ser escrito
20 Linha_end EQU 022FH ;Fim da linha onde acaba onde o
    alimento foi escrito
21
22 ;Constantes do programa e opções
23 caraterVazio EQU 20H ;Carater para limpar o ecrã
24
25 MBalance EQU 1 ;Opção para a escolha do Menu
    balança
26 MTotalequ 2 ;Opção para a escolha do Menu
    Total
27 MReset EQU 3 ;Opção para a escolha do Menu
    Reset
28
29 Base_Tabela EQU 6000H ;Inicio da tabela que contém os
    alimentos
30
31 ;Posição onde vai ser escrito no menu principal da balança o peso
32 LocalizacaoPesoDisplay EQU 0248H ;Endereço base do display onde
    ficará localizado o peso
33
34 ;Posição onde vai ser escrito no menu total as proteínas
35 LocalizacaoPDisplay EQU 0228H ;Endereço base do display onde
    ficará localizado o peso
36
37 ;Endereços na memória principal onde serão guardados os macronutrientes e calorias
38 LocalizacaoMemoriaP EQU 02A0H ;Endereço da memória principal
    onde irá ficar guardado o valor das proteínas totais
39 LocalizacaoMemoriaC EQU 02B0H ;Endereço da memória principal
    onde irá ficar guardado o valor dos hidratos de carbono totais
40 LocalizacaoMemoriaF EQU 02C0H ;Endereço da memória principal
    onde irá ficar guardado o valor das gorduras totais
41 LocalizacaoMemoriaCal EQU 02D0H ;Endereço da memória principal
    onde irá ficar guardado o valor das calorias totais
42
43 ;Endereços na memória secundária onde serão guardados os macronutrientes e calorias
44 LocalizacaoMemoriaPA EQU 02F0H ;Endereço da memória secundária
    onde irá ficar guardado o valor das proteínas totais
45 LocalizacaoMemoriaCA EQU 0300H ;Endereço da memória secundária
    onde irá ficar guardado o valor dos hidratos de carbono totais
46 LocalizacaoMemoriaFA EQU 0310H ;Endereço da memória secundária
    onde irá ficar guardado o valor das gorduras totais
47 LocalizacaoMemoriaCaLA EQU 0320H ;Endereço da memória secundária
    onde irá ficar guardado o valor das calorias totais
48
49 NR_TOT_CAR EQU 4 ;Número total de caracteres para
    a converção de hexadecimal para decimal
50
51 StackPointer EQU 7000H ;Endereço da pilha
52
53 ;-----
    -----
    -----

```

```

54 ;
55 menus
56 ;-----
57 ;*****
58 ;Menu Principal
59 ;*****
60 Place 2000H
61 MenuPrincipal: ;Display do menu principal
62     String " MENU PRINCIPAL " ;linha 1 do display do menu principal
63     String " " ;linha 2 do display do menu principal
64     String " 1- Balança " ;linha 3 do display do menu principal
65     String " 2- Total " ;linha 4 do display do menu principal
66     String " 3- Reset " ;linha 5 do display do menu principal
67     String " " ;linha 6 do display do menu principal
68     String " " ;linha 7 do display do menu principal
69
70 ;*****
71 ;Menu Principal da Balança
72 ;*****
73 Place 2080H
74 MenuBalancaPrincipal: ;Display do menu principal da balança
75     String " MENU BALANÇA " ;linha 1 do display do menu principal da
76     String "Alimento: " ;linha 2 do display do menu principal da
77     String " " ;linha 3 do display do menu principal da
78     String "Peso: " ;linha 4 do display do menu principal da
79     String " g " ;linha 5 do display do menu principal da
80     String "1) Registrar " ;linha 6 do display do menu principal da
81     String "OK para voltar " ;linha 7 do display do menu principal da
82
83 ;*****
84 ;Menus da seleção de alimento para a balança 1, 2, 3, 4 e 5
85 ;*****
86 Place 2100H
87 MenuBalanca1: ;Display do menu balança 1
88     String " MENU BALANÇA " ;linha 1 do display do menu balança 1
89     String "1 - Aveia " ;linha 2 do display do menu balança 1
90     String "2 - Pão de Forma" ;linha 3 do display do menu balança 1
91     String "3 - Batata " ;linha 4 do display do menu balança 1
92     String "4 - Arroz " ;linha 5 do display do menu balança 1
93     String "5 - Feijão " ;linha 6 do display do menu balança 1
94     String ">>>> Change>>>> " ;linha 7 do display do menu balança 1
95
96 Place 2180H
97 MenuBalanca2: ;Display do menu balança 2
98     String "6 - Legumes " ;linha 1 do display do menu balança 2
99     String "7 - Tomate " ;linha 2 do display do menu balança 2
100     String "8 - Banana " ;linha 3 do display do menu balança 2
101     String "9 - Laranja " ;linha 4 do display do menu balança 2
102     String "10- Maça " ;linha 5 do display do menu balança 2
103     String "11- Kiwi " ;linha 6 do display do menu balança 2
104     String ">>>> Change>>>> " ;linha 7 do display do menu balança 2
105
106 Place 2200H
107 MenuBalanca3: ;Display do menu balança 3
108     String "12- Bolacha Choc" ;linha 1 do display do menu balança 3
109     String "13- Pizza Q&F PD" ;linha 2 do display do menu balança 3
110     String "14- Amêndoas " ;linha 3 do display do menu balança 3
111     String "15- Linhaça " ;linha 4 do display do menu balança 3
112     String "16- Azeite " ;linha 5 do display do menu balança 3
113     String "17- Leite Magro " ;linha 6 do display do menu balança 3
114     String ">>>> Change>>>> " ;linha 7 do display do menu balança 3
115
116 Place 2280H

```

```

117 MenuBalanca4:                                ;Display do menu balança 4
118     String "18- Whey                        " ;linha 1 do Display do menu balança 4
119     String "19- Salmão                      " ;linha 2 do Display do menu balança 4
120     String "20- Pescada                     " ;linha 3 do Display do menu balança 4
121     String "21- Atum                        " ;linha 4 do Display do menu balança 4
122     String "22- Porco                       " ;linha 5 do Display do menu balança 4
123     String "23-Frango(peito)"              ;linha 6 do Display do menu balança 4
124     String ">>>> Change>>>>"              ;linha 7 do Display do menu balança 4
125
126 Place 2300H
127 MenuBalanca5:                                ;Display do menu balança 5
128     String "24- Peru (peito)"              ;linha 1 do display do menu balança 5
129     String "25- Ovo                         " ;linha 2 do display do menu balança 5
130     String "26- Queijo                     " ;linha 3 do display do menu balança 5
131     String ">>>> Change>>>>"              ;linha 4 do display do menu balança 5
132     String "                                " ;linha 5 do display do menu balança 5
133     String " OK para voltar "              ;linha 6 do display do menu balança 5
134     String " menu principal "              ;linha 7 do display do menu balança 5
135
136 ;*****
137 ;Menu Opção Errada
138 ;*****
139 Place 2380H
140 MenuErro:                                    ;Display do menu erro
141     String "      ATENÇÃO      "           ;linha 1 do display do menu erro
142     String "                   "           ;linha 2 do display do menu erro
143     String "      OPÇÃO       "           ;linha 3 do display do menu erro
144     String "      ERRADA      "           ;linha 4 do display do menu erro
145     String "                   "           ;linha 5 do display do menu erro
146     String " OK para voltar "           ;linha 6 do display do menu erro
147     String "                   "           ;linha 7 do display do menu erro
148
149 ;*****
150 ;Menu Total - Macronutrientes
151 ;*****
152 Place 2400H
153 MenuTotal:                                  ;Display do menu total de macronutrientes
154     String "      MENU TOTAL  "           ;linha 1 do display do menu total de
155     String "                   "           ;linha 2 do display do menu total de
156     String " P:                g "         ;linha 3 do display do menu total de
157     String " C:                g "         ;linha 4 do display do menu total de
158     String " F:                g "         ;linha 5 do display do menu total de
159     String " Cal:              K "         ;linha 6 do display do menu total de
160     String " OK para voltar "           ;linha 7 do display do menu total de
161     String " macronutrientes "
162
163 ;*****
164 ;Menu Reset
165 ;*****
166 Place 2480H
167 MenuReset:                                  ;Display do menu reset
168     String "      MENU RESET  "           ;linha 1 do display do menu reset
169     String "                   "           ;linha 2 do display do menu reset
170     String " 1) Confirmar     "           ;linha 3 do display do menu reset
171     String "                   "           ;linha 4 do display do menu reset
172     String "                   "           ;linha 5 do display do menu reset
173     String "OK para cancelar"           ;linha 6 do display do menu reset
174     String "      e voltar    "           ;linha 7 do display do menu reset
175
176 ;*****
177 ;Menu Erro Overflow
178 ;*****
179 Place 2500H
180 MenuErroOverflow:                          ;Display do menu erro de overflow
181     String "      ATENÇÃO      "           ;linha 1 do display do menu erro
182     String "                   "           ;linha 2 do display do menu erro
183     String " Ocorre overflow"           ;linha 3 do display do menu erro

```

```

183 String " " ;linha 4 do display do menu erro
184 String " " ;linha 5 do display do menu erro
185 String " " ;linha 6 do display do menu erro
186 String " OK para voltar " ;linha 7 do display do menu erro
187
188 ;*****
189 ;Menu Reset após a confirmação do RESET
190 ;*****
191 Place 2580H
192 MenuResetConfirmado: ;Display do menu reset
193 String " MENU RESET " ;linha 1 do display do menu reset
194 String " " ;linha 2 do display do menu reset
195 String " Limpeza total " ;linha 3 do display do menu reset
196 String " com sucesso! " ;linha 4 do display do menu reset
197 String " " ;linha 5 do display do menu reset
198 String " OK para voltar " ;linha 6 do display do menu reset
199 String " menu principal " ;linha 7 do display do menu reset
200
201 ;-----
202 ;
203 Principais ;-----
204
205 ;*****
206 ;Programa Principal
207 ;*****
208
209 Place 0000H
210 Incio: ;Inicio do Programa
211 MOV R0, ProgramaPrincipal ;Coloca em R0 o endereço da tag
212 ProgramaPrincipal
213 JMP R0 ;Salta para tag ProgramaPrincipal
214 Place 3000H ;Início onde o código ficará guardado na
memória
215
216 ProgramaPrincipal:
217 MOV SP, StackPointer ;Colocar o endereço da pilha na SP
218 CALL LimpaDisplay ;Chamada a função limpa display, que faz
limpeza do display
219 CALL LimpaPerifericos ;Chamada a função limpa periféricos para
limpar os periféricos de entrada
220
221 Liga:
222 MOV R0, ON_OFF ;Coloca no registo R0 o valor atual do
botão ON/OFF
223 MOVB R1, [R0] ;Passamos o valor guardado no byte de R0
e guardamos no registo R1
224 CMP R1, 1 ;Verifica se o botão está ligado
225 JNE Liga ;Se continuar a zero, então volta a
verificar o estado atual até ser ligado a balança
226
227 Ligado:
228 MOV R2, MenuPrincipal ;Coloca em R2 o endereço do
"MenuPrincipal"
229 CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuPrincipal"
230 CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
231
232 Le_Opcao:
233 CALL RDesliga ;Chamada a rotina "RDesliga" , a
qualquer a balança pode ser desligada
234 MOV R0, SEL_NR_MENU ;Coloca no registo R0 o endereço do
periférico de entrada "SEL_NR_MENU"
235 MOVB R1, [R0] ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "SEL_NR_MENU"
236 CMP R1, 0 ;Compara o valor do registo R1 com
constante zero

```

```

237 JEQ Le_Opcao ;Caso sejam iguais continua o lendo a
opção que está a ser colocada no periférico "SEL_NR_MENU"
238 CMP R1, MBalanca ;Compara o valor do registo R1 com o
valor da constante "MBalanca"
239 JEQ MeBalanca ;Caso ambos os valores sejam iguais
então salta para o endereço "MeBalanca", assim podendo escolher um alimento
240 CMP R1, MTTotal ;Compara o valor do registo R1 com o
valor da constante "MTTotal"
241 JEQ MeTotal ;Caso ambos os valores sejam iguais
então salta para o endereço "MeTotal", assim podendo visualizar os total de
macronutrientes e calorias
242 CMP R1, MReset ;Compara o valor do registo R1 com o
valor da constante "MReset"
243 JEQ MeReset ;Caso ambos os valores sejam iguais
então salta para o endereço "MeReset", assim podendo dar reset ao total de
macronutrientes e calorias ou cancelar
244 ;Caso nenhuma das opções acima tenha
sido escolhida então aparece uma
mensagem de erro
245 MOV R2, MenuErro ;Coloca em R2 o endereço do "MenuErro"
246 CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuErro"
247 CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
248 CALL RErro ;Chamada a rotina "RErro" para validar o
OK (periférico "OK") do "MenuErro"
249
250 ;*****
251 ;Menu Total
252 ;*****
253 MeTotal:
254 MOV R2, MenuTotal ;Coloca em R2 o endereço do "MenuTotal"
255 CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuTotal"
256 CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
257 MOV R0, OK ;Coloca no registo R0 o endereço do
periférico de entrada "OK"
258 Ciclo_menutotal:
259 CALL RDesliga ;Chamada a rotina "RDesliga" , a
qualquer a balança pode ser desligada
260 CALL Converte_to_CHAR_M ;Chamada a rotina "Converte_to_CHAR"
para converter o valor dos macronutrientes em hexadecimal em char
261 MOVB R1, [R0] ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "OK"
262 CMP R1 , 1 ;Compara o valor do registo R1 com
constante 1
263 JNE Ciclo_menutotal ;Caso sejam diferentes, salta para a tag
"Ciclo_menutotal"
264 JMP Ligado ;Caso sejam iguais então salta para tag
"Ligado", significa que o utilizador colocou "OK" a 1, após
265 ;ter visualizado os totais de
macronutrientes e calorias regressou
para o menu principal
266 ;*****
267 ;Menu Reset
268 ;*****
269 MeReset:
270 MOV R2, MenuReset ;Coloca em R2 o endereço do "MenuReset"
271 CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuReset"
272 CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
273 MOV R3,SEL_NR_MENU ;Coloca no registo R3 o endereço do
periférico de entrada "SEL_NR_MENU"
274 MOV R0, OK ;Coloca no registo R0 o endereço do
periférico de entrada "OK"
275
276 Ciclo_menureset:
277 CALL RDesliga ;Chamada a rotina "RDesliga" , a
qualquer a balança pode ser desligada
278 MOVB R1, [R0] ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "OK"

```

```

279     MOV B R4, [R3] ;Coloca no registo R4 o conteúdo do
endereço do periférico de entrada "SEL_NR_MENU"
280     CMP R4 ,1 ;Compara o valor do registo R4 com
constante 1
281     JEQ Reset_memoria ;Caso ambos os valores sejam iguais
então salta para a tag "Reset_memoria"
282     CMP R1, 1 ;Compara o valor do registo R1 com
constante 1
283     JNE Ciclo_menureset ;Caso sejam diferentes, salta para a tag
"Ciclo_menureset"
284     JMP Ligado ;Caso ambos os valores sejam iguais,
salta para tag "Ligado", o que significa que cancelou o reset
285 ;ao total do macronutrientes e calorias
286
287 Reset_memoria:
288     MOV R2, MenuResetConfirmado ;Coloca em R2 o endereço do
"MenuResetConfirmado"
289     CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuResetConfirmado"
290     CALL LimpaMacronutrientes ;Chamada a rotina "LimpaMacronutrientes"
para limpar os totais de P, C ,F e Cal
291     CALL LimpaMemoriaSecundaria ;Chamada a rotina
"LimpaMemoriaSecundaria" para limpar valores de P, C ,F e Cal na memória
secundária, significando que havendo overflow, os valores não somados ao total
dos macronutrientes e calorias
292     JMP Verificacao ;Salta para tag "Verificacao", que
verifica o "OK" para voltar ao menu principal
293
294 ;*****
295 ;Menu Balança
296 ;*****
297 MeBalanca:
298     MOV R2, MenuBalanca1 ;Coloca em R2 o endereço do "MenuBalanca1"
299     CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuBalanca1"
300     CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
301     MOV R0, CHANGE ;Coloca no registo R0 o endereço do
periférico de entrada "CHANGE"
302
303 Ciclo_menubalanca:
304     CALL RDesliga ;Chamada a rotina "RDesliga" , a
qualquer a balança pode ser desligada
305     MOV B R1, [R0] ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "CHANGE"
306     CMP R1, 1 ;Compara o registo R1 com a constante 1
307     JEQ Mudar1 ;Caso ambos os valores sejam iguais
então para o tag "Mudar1", que sigifica que o botão "CHANGE" é 1, assim mudando
para o menu "MenuBalanca2"
308 ;Caso sejam diferente então irá para
rotina "Verifica_selecao" para ver se
algum dos alimentos foi selecionado
309     CALL Verifica_selecao ;Chamada a rotina "Verifica_selecao",
verificando se algum alimento foi selecionado
310     JMP Ciclo_menubalanca ;Caso contrário, permace no ciclo
saltando para a tag "Ciclo_menubalanca"
311
312 Mudar1:
313     MOV R2, MenuBalanca2 ;Coloca em R2 o endereço do "MenuBalanca2"
314     CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuBalanca2"
315     CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
316     MOV R0, CHANGE ;Coloca no registo R0 o endereço do
periférico de entrada "CHANGE"
317
318 Ciclo_menubalanca1:
319     CALL RDesliga ;Chamada a rotina "RDesliga" , a
qualquer a balança pode ser desligada
320     MOV B R1, [R0] ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "CHANGE"
321     CMP R1, 1 ;Compara o registo R1 com a constante 1
322     JEQ Mudar2 ;Caso ambos os valores sejam iguais

```



```

qualquer a balança pode ser desligada
365 MOV B R1, [R0] ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "CHANGE"
366 CMP R1, 1 ;Compara o registo R1 com a constante 1
367 JEQ MeBalanca ;Caso ambos os valores sejam iguais
então para o tag "MeBalanca, que significa que o botão "CHANGE" é 1, assim
voltando para o início, mudando para o menu "MeBalanca"
368 ;Caso sejam diferente então irá para
rotina "Verifica_selecao" para ver se
algum dos alimentos foi selecionado
369
370 CALL Verifica_selecao ;Chamada a rotina "Verifica_selecao",
verificando se algum alimento foi selecionado
371 MOV R3, OK ;Coloca no registo R3 o endereço do
periférico de entrada "OK"
372 MOV B R3, [R3] ;Coloca no registo R3 o conteúdo do
endereço do periférico de entrada "OK"
373 CMP R3, 1 ;Compara o registo R3 com a constante 1
374 JNE Ciclo_menubalanca4 ;Caso os valores não sejam iguais
permane no ciclo saltando para a tag "Ciclo_menubalanca4"
375 JMP Ligado ;Caso os valores sejam iguais então
salta para tag "Ligado" , significa que o "OK" está ativo então volta para o menu
principal do programa
376
377 ;*****
378 ;Menu Balança Principal
379 ;*****
380
381 MeBalancaPrincipal:
382 CALL Validacao ;Chamada a rotina "Validacao" , porque
após selecionar o alimento é necessário validar com "OK" para passar ao
"MenuBalancaPrincipal"
383 MOV R2, MenuBalancaPrincipal ;Coloca em R2 o endereço do
"MenuBalancaPrincipal"
384 CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuBalancaPrincipal"
385 CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
386 MOV R0 , Linha_inic ;Coloca no registo R0 o valor da
constante "Linha_inic", que contém endereço do display onde deve começar a ser
escrito o nome do alimento selecionado
387 MOV R1, Linha_end ;Coloca no registo R1 o valor da
constante "Linha_end", que contém endereço do display onde deve terminar a ser
escrito o nome do alimento selecionado
388
389 Ciclo_MostraLinha:
390 MOV B R4, [R9] ;Coloca em R4 o conteúdo no endereço do
alimento selecionado
391 MOV B [R0], R4 ;Coloca na posição de memória R0
(colando o primeiro byte que aparece na String do nome do alimento selecionado)
o valor do registo R4
392 ADD R9, 1 ;Incrementa 1 ao registo R9, assim
andando no nome um byte, para ser possível ser escrito o próximo char que aparece
393 ADD R0, 1 ;Adiciona ao registo a constante 1,
para ser possível ser adicionado o seguinte char até o fim da linha do display
394 CMP R0, R1 ;Compara R0 e R1
395 JLE Ciclo_MostraLinha ;Enquanto o R0 for menor ou igual ao
registo R1 salta para a tag "Ciclo_MostraLinha", pois caso contrário já foi
preenchida a linha inteira do display
396 ;com o nome do alimento selecionado e já
é possível inserir o peso
397
398 PesoInserido:
399 CALL RDesliga ;Chamada a rotina "RDesliga" , a
qualquer a balança pode ser desligada
400 MOV R6, PESO ;Coloca no registo R6 o endereço do
periférico de entrada "PESO"
401 MOV R8, [R6] ;Coloca no registo R8 conteúdo do
endereço do periférico de entrada "PESO"
402 MOV R7, 3000 ;Coloca no registo R7 a constante 3000
(peso máximo que pode ser inserido na balança em gramas)
403 CMP R8, R7 ;Compara o valor do registo R8 com o
valor do registo R7
404 JLE Converte ;Caso o valor do registo R8 (peso) seja

```



```

menor ou igual que o valor do R7 (3000), salta para tag "Converte", significa
que o peso foi aceite
404 MOV R8, 0 ;Caso o valor do registo tenha sido
maior que 3000, então coloca no registo R8 a constante 0
405 MOV [R7] , R8 ;Coloca na posição de memória R7
(colocando zero, assim significa que o peso ultrapassou os 3000 e fica zero no
display) o valor do registo R8
406
407 Converte:
408 CALL Converte_to_CHAR ;Chamada a rotina "Converte_to_CHAR"
para converter o valor inserido em hexadecimal em char
409 MOV R0, SEL_NR_MENU ;Coloca no registo R6 o endereço do
periférico de entrada "SEL_NR_MENU"
410 MOVB R1, [R0] ;Coloca em R1 o conteúdo do endereço do
periférico de entrada "SEL_NR_MENU"
411 MOV R3, 1 ;Coloca em R3 a constante 1
412 CMP R1, R3 ;Compara o valor do registo R1 com o
valor do R3 (para ver se utilizador confirmou que queria registar o alimento)
413 JNE Verifica_OK ;Caso os valores sejam diferentes então
salta para tag "Verifica_OK", ficando espera que o utilizador sai do menu
balança principal, podendo sempre alterar o peso a ser inserido
414 MOV R10, [R6] ;Caso os valores sejam iguais significa
que foi registado o alimento, então no registo R10 é colocado o conteúdo de R6
(peso inserido)
415 CALL CalculadoraMacros ;Chamada a rotina "CalculadoraMacros"
para calcular os macronutrientes ficando guardado na memória secundária os
cálculos
416 CMP R8, 1 ;No fim da rotina o registo R8 contém o
valor de 0 ou 1 (0 caso não houver ocorrência de overflow ou 1 caso houver)
assim comparando o valor do registo R8 a constante 1
417 JEQ ErroExcessoOver ;Caso os valores sejam iguais salta para
a tag "ErroExcessoOver"
418 CALL CalculadoraCalorias ;Chamada a rotina "CalculadoraCalorias"
para calcular a quantidade de calorias ficando guardado na memória secundária os
cálculos
419 CMP R8, 1 ;No fim da rotina o registo R8 contém o
valor de 0 ou 1 (0 caso não houver ocorrência de overflow ou 1 caso houver)
assim comparando o valor do registo R8 a constante 1
420 JEQ ErroExcessoOver ;Caso os valores sejam iguais salta para
a tag "ErroExcessoOver"
421 CALL MemoriaMacroSecundaria ;Chamada a rotina
"MemoriaMacroSecundaria" para adicionar o que está guardado na memória
secundária com a memória principal
422 CMP R8, 1 ;No fim da rotina o registo R8 contém o
valor de 0 ou 1 (0 caso não houver ocorrência de overflow ou 1 caso houver)
assim comparando o valor do registo R8 a constante 1
423 JEQ ErroExcessoOver ;Caso os valores sejam iguais salta para
a tag "ErroExcessoOver"
424 CALL MemoriaMacro ;Caso não haja overflow é chamada a
rotina "MemoriaMacro", que serve para passar o valores que estão na memória
secundária para a memória principal onde estão os valores definitivos que
425 ;serão apresentados no menu total
426 CALL LimpaMemoriaSecundaria
427
428 Verifica_OK:
429 MOV R0, OK ;Coloca no registo R0 o endereço do
periférico de entrada "OK"
430 MOVB R1, [R0] ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "OK"
431 CMP R1 ,1 ;Compara R1 com a constante 1
432 JNE PesoInserido ;Caso os valores sejam diferentes salta
para tag "PesoInserido", significando que o utilizador ainda não deu "OK" para
voltar para o "MenuPrincipal"
433 JMP Ligado ;Caso os valores sejam iguais então
salta para tag "Ligado" , significa que o "OK" está ativo então volta para o
"MenuPrincipal"
434
435 ErroExcessoOver:
436 MOV R2, MenuErroOverflow ;Coloca em R2 o endereço do
"MenuErroOverflow"
437 CALL MostraDisplay ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuErroOverflow"
438 CALL LimpaPerifericos ;Chamada a rotina "LimpaPerifericos"

```

```

439     para limpar os periféricos de entrada
CALL LimpaMemoriaSecundaria           ;Chamada a rotina
"LimpaMemoriaSecundaria" para limpar valores de P, C ,F e Cal na memória
secundária, significando que havendo overflow, os valores não somados ao total
dos macronutrientes e calorias

440
441 Verificacao:
442     MOV R0, OK                       ;Coloca no registo R0 o endereço do
periférico de entrada "OK"
443     MOVB R1, [R0]                   ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "OK"
444     CMP R1 ,1                       ;Compara R1 com a constante 1
445     JNE Verificacao                 ;Caso os valores não sejam iguais, então
volta para a tag "Verificacao"
446     JMP Ligado                      ;Caso os valores sejam iguais então
salta para tag "Ligado" , significa que o "OK" está ativo então volta para o menu
principal do programa

447
448
449 ;-----
-----
450 ;                                     Rotinas
451 ;-----
-----
452 ;*****
453 ;Rotina Verifica Seleccion
454 ;*****
455
456 Verifica_selecao:
457     PUSH R4
458     ;-----
PUSH R5                               ;Guarda na pilha os registos que são
alterados na rotina
459     PUSH R6
460     PUSH R7
461     ;-----
MOV R4, SEL_NR_MENU                   ;Coloca no registo R4 o endereço do
periférico de entrada "SEL_NR_MENU"
462     MOVB R5, [R4]                   ;Coloca no registo R5 o conteúdo do
endereço do periférico de entrada "SEL_NR_MENU"
463     MOV R9, R5                       ;Coloca no registo R9 o valor do registo
R5 (o numero do alimento seleccionado)
464     MOV R6, 1AH                      ;Coloca no registo R6 a constante 1AH
(26 em decimal pois a selecção do alimento vai só até 26)
465     CMP R9, R6                       ;Compara o valor do registo R9 com o
valor do registo R6
466     JLE Opcao_certa                 ;Caso o valor de R9 seja menor ou igual
a R6, significa que a opção é certa
467     MOV R2, MenuErro                 ;Coloca em R2 o endereço do
"MenuBalancaPrincipal"
468     CALL MostraDisplay               ;Chamada a rotina "MostraDisplay" para
mostrar no display o "MenuErro"
469     CALL LimpaPerifericos           ;Chamada a rotina "LimpaPerifericos"
para limpar os periféricos de entrada
470     CALL RErro                      ;Caso o valor de R9 tenha sido maior que
o valor R6, é feita a chamada da rotina "RErro" que mostra uma mensagem a dizer
que foi seleccionado
471                                     ;um alimento não válido
472 Opcao_certa:
473     SUB R9, 1                         ;Subtrai ao registo R9 a constante 1
(para que seja possível aceder colocamos a base da tabela onde contém o primeiro
elemento)
474     MOV R7 , 20H                     ;Coloca em R7 a constante 20H
475     MUL R9, R7                       ;Multiplica o valor do registo R9 com o
valor do registo R7, pois as Strings com os nomes dos alimentos é de 20H em 20H
na memória
476     MOV R7, Base_Tabela              ;Coloca em R7 o endereço da constante
"Base_Tabela" (onde começa na memória a tabela)
477     ADD R9, R7                       ;Adiciona o valor do registo R9 com o
valor do registo R7
478     CMP R5, 0                       ;Compara o valor do R5 com a constante 0

```

```

479     JNZ MeBalancaPrincipal                ;Caso não seja zero, então salta para a
tag "MeBalancaPrincipal" onde estará a balança com o nome do alimento escolhido
480                                           ;Compara com zero o registo R5 pois
apesar do número ser menor que 26,
481     POP R7                                ;Retira da pilha os registos guardados
;-----
482     POP R6                                ;
483     POP R5                                ;
484     POP R4                                ;
;-----
485     RET
486
487 ;*****
488 ;Rotina Valida seleção
489 ;*****
490
491 Validacao:
492     PUSH R0                                ;Guarda na pilha os registos que são
alterados na rotina
493     PUSH R1                                ;
;-----
494     MOV R0, OK                            ;Coloca no registo R0 o endereço do
periférico de entrada "OK"
495
496 Ciclo_V:
497     MOVB R1, [R0]                        ;Coloca no registo R1 o conteúdo do
endereço do periférico de entrada "OK"
498     CMP R1, 1                            ;Compara R1 com a constante 1
499     JEQ Fim_Validacao                    ;Caso os valores sejam iguais, então
salta para a tag "Fim_Rotina", pois para selecionar o alimento é necessário dar
"OK" para validar essa seleção
500     JMP Ciclo_V                          ;Caso não sejam iguais, então salta para
a tag Ciclo_V
501
502 Fim_Validacao:
503     POP R1                                ;
;-----
504     POP R0                                ;Retira da pilha os registos guardados
505     RET
506
507 ;*****
508 ;Rotina Calcula Macronutrientes
509 ;*****
510 CalculadoraMacros:
511     PUSH R0                                ;
;-----
512     PUSH R1                                ;
513     PUSH R2                                ;
514     PUSH R3                                ;
515     PUSH R4                                ;Guarda na pilha os registos que são
alterados na rotina
516     PUSH R5                                ;
517     PUSH R6                                ;
518     PUSH R7                                ;
519     PUSH R9                                ;
;-----
520     MOV R0, SEL_NR_MENU                    ;Coloca no registo R0 o endereço do
periférico de entrada "SEL_NR_MENU"
521     MOV R5, 0                            ;Coloca em R5 o valor da constante 0
522     MOVB [R0], R5                        ;Coloca na posição de memória R0
(colocando zero) o valor do registo R5
523
524     MOV R8, 0                            ;Coloca em R8 a constante 0 , para
indicação se ocorreu ou não overflow no fim da rotina, pois caso este registo
não mude para 1 no fim da rotina, então não ocorre overflow
525     MOV R0, R9                            ;Coloca em R0 o valor de R9, assim
passando o endereço do alimento que foi selecionado
526     MOV R2, 10H                          ;Coloca em R2 a constante 10H (16 em
decimal), pois os macronutrientes estão mais 10H da posição de memória do nome
de cada um dos alimentos
527     MOV R3, 64H                          ;Coloca em R3 a constante 64H (100 em
decimal)

```

```

528     MOV R4, 3                                ;Coloca em R4 a constante 3 (contador)
529     MOV R5, LocalizacaoMemoriaPA            ;Coloca em R5 o endereço da constante
"LocalizacaoMemoriaPA", esta constante contém a localização na memória
secundária onde vai ficar guardado as proteínas totais

530
531 Cont_Calculadora:
532     MOV R1, R10                              ;Coloca em R1 o valor de R10 que é o
valor do peso que foi inserido
533     MOVB R7, [R0]                            ;Coloca em R7 o conteúdo do registo R0 (
que contém as proteínas do alimento foi selecionado )
534     MUL R1, R7                              ;Multiplica o valor R1 com o valor R7
(multiplicando o peso pelas proteínas do alimento selecionado)
535     JV ErroOverflow                          ;Salta caso haja overflow para a tag
"ErroOverflow"
536     MOV R6, R1                              ;Coloca em R6 o valor de R1 (resultado
da multiplicação do peso pelas proteínas para que não se perca esse valor em R1)
537     DIV R1, R3                              ;Divide o valor do R1 com o valor de R3
, pois divide-se o resultado da multiplicação por 100 devido a que as proteínas
estão por cada 100g de peso do alimento)
538     MOD R6, R3                              ;Resto da divisão inteira do valor do
resgisto R6 com o valor do registo R3 (necessário para os arredondamentos)
539     MOV R9, 32H                             ;Coloca em R9 a constante 32H (50 em
decimal)
540     CMP R6, R9                              ;Compara valor de R6 com o valor de R9
pois é necessário ver se o resto da divisão deu menor que 50 para arredondar
para cima ou para baixo
541     JLT Continua                            ;Caso o valor de R6 seja menor que o
valor em R9, significa que foi menor que 50 então salta para a tag "Continua"
mantendo o valor que deu na divisão inteira
542     ADD R1, 1                              ;Caso o valor de R6 seja maior ou igual
ao valor de R9 então é necessário somar 1 ao registo R1, que tem o valor da
divisão assim havendo arredondamento

543
544 Continua:
545     MOV [R5], R1                            ;Coloca na posição de memória R5
(guardando assim o valor das proteínas na memória secundária) o valor do registo
R1
546     ADD R5, R2                              ;Adiciona ao valor do registo R5 o valor
do registo R2, assim saltando na memória 10H para guardar o macronutriente
seguinte
547     ADD R0, 1                              ;Adiciona o valor do registo R0 a
constante 1 (fazendo com que na tabela esteja a ser percorrido o macronutriente
seguinte)
548     SUB R4, 1                              ;Subtrai o valor do registo R4 a
constante 1 (para saber que um dos 3 macronutrientes já foi calculado)
549     CMP R4, 0                              ;Compara o valor do registo R4 com a
constante 0
550     JEQ Fim_Calculadora                    ;Caso sejam iguais, então significa que o
contador chegou ao fim e já foi calculado os macronutrientes (proteínas,
hidratos de carbono e gorduras) e salta para tag "Fim_Calculadora"
551     JMP Cont_Calculadora                    ;Caso sejam diferentes significa que irá
calcular o macronutriente seguinte saltando para a tag "Cont_Calculadora"

552
553 ErroOverflow:
554     MOV R8, 1                              ;Coloca em R8 a constante 1,
significando que houve ocorrencia de overflow

555
556 Fim_Calculadora:
557     POP R9
;-----
558     POP R7                                  ;
559     POP R6                                  ;
560     POP R5                                  ;
561     POP R4                                  ;Retira da pilha os registos guardados
562     POP R3                                  ;
563     POP R2                                  ;
564     POP R1                                  ;
565     POP R0                                  ;
;-----
566     RET
567
568 ;*****
569 ;Rotina Calcula Calorias

```

```

570 ;*****
571 CalculadoraCalorias:
572     PUSH R0
573     ;-----
574     PUSH R1 ;
575     PUSH R2 ;
576     PUSH R3 ;Guarda na pilha os registos que são
577     alterados na rotina
578     PUSH R4 ;
579     PUSH R5
580     ;-----
581     MOV R8, 0 ;Coloca em R8 a constante 0 , para
582     indicação se ocorreu ou não overflow no fim da rotina, pois caso este registo
583     não mude para 1 no fim da rotina, então não ocorre overflow
584     MOV R0, LocalizacaoMemoriaPA ;Coloca em R0 o endereço da constante
585     "LocalizacaoMemoriaPA" , esta constante contém a localização na memória
586     secundária onde estão guardado as proteínas calculadas
587     MOV R1, [R0] ;Coloca o conteúdo do registo R0 no
588     registo R1 (contém o número de proteínas calculadas na memória secundária)
589     MOV R2, LocalizacaoMemoriaCA ;Coloca em R2 o endereço da constante
590     "LocalizacaoMemoriaCA" , esta constante contém a localização na memória
591     secundária onde estão guardado os hidratos de carbono calculadas
592     MOV R3, [R2] ;Coloca o conteúdo do registo R2 no
593     registo R3 (contém o número de hidratos de carbonos calculadas na memória
594     secundária)
595     MOV R4, LocalizacaoMemoriaFA ;Coloca em R4 o endereço da constante
596     "LocalizacaoMemoriaF" , esta constante contém a localização na memória
597     secundária onde estão guardado as gorduras calculadas
598     MOV R5, [R4] ;Coloca o conteúdo do registo R4 no
599     registo R5 (contém o número de gorduras calculadas na memória secundária)
600     MOV R6, 4 ;Coloca em R6 a constante 4 (necessária
601     para a multiplicação do calculo das calorias)
602     MOV R7, 9 ;Coloca em R7 a constante 9 (necessária
603     para a multiplicação do calculo das calorias)
604
605 Calcula_Calorias:
606     ADD R1, R3 ;Adiciona o valor do resgisto R1 e o
607     valor R3 ( pois as proteínas e os hidratos de carbono vão ser ambos
608     multiplicados por 4, facilitando as contas)
609     MUL R1, R6 ;Multiplica o valor de R1 (resultado da
610     soma) com o valor de R6 (constante 4)
611     JV ErroOverflowCalorias ;Salta caso haja overflow para a tag
612     "ErroOverflowCalorias"
613     MUL R5, R7 ;Multica o valor de R5 com o valor de R7
614     (multiplicando as gorduras por 9)
615     JV ErroOverflowCalorias ;Salta caso haja overflow para a tag
616     "ErroOverflowCalorias"
617     ADD R1, R5 ;Adiciona o valor de R1 o valor de R5 (o
618     resultado terá (P + C) * 4 + F * 9)
619     MOV R3, LocalizacaoMemoriaCalA ;Coloca em R3 o endereço da constante
620     "LocalizacaoMemoriaCalA" , esta constante contém a localização na memória
621     secundária onde vai ficar guardado as calorias calculadas
622     MOV [R3], R1 ;Coloca na posição de memória R3
623     (guardando assim o valor das calorias na memória secundária) o valor do registo R1
624     JMP Fim_Calculadora_Calorias
625
626 ErroOverflowCalorias:
627     MOV R8, 1 ;Coloca em R8 a constante 1,
628     significando que houve ocorrencia de overflow
629
630 Fim_Calculadora_Calorias:
631     POP R5
632     ;-----
633     POP R4 ;
634     POP R3 ;Retira da pilha os registos guardados
635     POP R2 ;
636     POP R1 ;
637     POP R0 ;
638     ;-----
639     RET
640
641
642

```

```

613 ;*****
614 ;Rotina MemoriaMacroSecundaria
615 ;*****
616 MemoriaMacroSecundaria:
617     PUSH R0
        ;-----
618     PUSH R1 ;
619     PUSH R2 ;
620     PUSH R3 ;
621     PUSH R4 ;Guarda na pilha os registos que são
alterados na rotina
622     PUSH R5 ;
623     PUSH R6
        ;-----
624
625     MOV R8, 0 ;Coloca em R8 a constante 0 , para
        indicação se ocorreu ou não overflow no fim da rotina, pois caso este registo
        não mude para 1 no fim da rotina, então não ocorre overflow
626     MOV R0, LocalizacaoMemoriaP ;Coloca em R0 o endereço da constante
        "LocalizacaoMemoriaP" , esta constante contém a localização na memória principal
        onde estão guardado as proteínas totais
627     MOV R1, LocalizacaoMemoriaPA ;Coloca em R1 o endereço da constante
        "LocalizacaoMemoriaPA" , esta constante contém a localização na memória
        secundária onde estão guardado as proteínas calculadas
628     MOV R2, 10H ;Coloca em R2 a constante 10H (16 em
        decimal), pois os macronutrientes estão mais 10H da posição de memória do nome
        de cada um dos alimentos
629     MOV R5, 4 ;Coloca em R5 a constante 4 (contador)
630
631 Ciclo_soma:
632     MOV R3, [R0] ;Coloca o conteúdo do registo R0 no
        registo R3 (contém o número de proteínas totais na memória principal)
633     MOV R4, [R1] ;Coloca o conteúdo do registo R1 no
        registo R4 (contém o número de proteínas calculadas na memória secundária)
634     ADD R3, R4 ;Adiciona ao valor do registo R3 o valor
        do registo R4 (adiciona o valor das proteínas totais com o valor das proteínas
        calculadas)
635     JV ErroOverflowSoma ;Salta caso haja overflow para a tag
        "ErroOverflowSoma"
636     ADD R0, R2 ;Adiciona ao valor do registo R0 o valor
        do registo R2, assim saltando na memória 10H para localização na memoria
        principal do macronutriente seguinte
637     ADD R1, R2 ;Adiciona ao valor do registo R1 o valor
        do registo R2, assim saltando na memória 10H para localização na memoria
        secundária do macronutriente seguinte
638     SUB R5, 1 ;Subtrai o valor do registo R5 a
        constante 1 (para saber que já percorreu os macronutrientes e calorias)
639     CMP R5, 0 ;Compara o valor do registo R5 com a
        constante 0
640     JNE Ciclo_soma ;Caso sejam diferentes volta para a tag
        "Ciclo_soma"
641     JMP Fim_rotina ;Caso sejam iguais, então significa que o
        contador chegou ao fim e já foi adicionado os macronutrientes (proteínas,
        hidratos de carbono e gorduras) e calorias entre a
642 ;a memória principal e secundária, assim
        saltando para tag "Fim_rotina"
643
644 ErroOverflowSoma:
645     MOV R8, 1 ;Coloca em R8 a constante 1,
        significando que houve ocorrencia de overflow
646
647 Fim_rotina:
648     POP R6
        ;-----
649     POP R5 ;
650     POP R4 ;Retira da pilha os registos guardados
651     POP R3 ;
652     POP R2 ;
653     POP R1 ;
654     POP R0
        ;-----
655     RET
656

```

```

657
658
659 ;*****
660 ;Menu MemoriaMacro
661 ;*****
662 MemoriaMacro:
663     PUSH R0
664     ;-----
665     PUSH R1 ;
666     PUSH R2 ;
667     PUSH R3 ;
668     PUSH R4 ;Guarda na pilha os registos que são
        alterados na rotina
669     PUSH R5 ;
670     PUSH R6
        ;-----
671     MOV R0, LocalizacaoMemoriaP ;Coloca em R0 o endereço da constante
        "LocalizacaoMemoriaP" , esta constante contém a localização na memória principal
        onde estão guardado as proteínas totais
672     MOV R1, LocalizacaoMemoriaPA ;Coloca em R1 o endereço da constante
        "LocalizacaoMemoriaPA" , esta constante contém a localização na memória
        secundária onde estão guardado as proteínas calculadas
673     MOV R2, 10H ;Coloca em R2 a constante 10H (16 em
        decimal), pois os macronutrientes estão mais 10H da posição de memória do nome
        de cada um dos alimentos
674     MOV R5, 4 ;Coloca em R5 a constante 4 (contador)
675
676 Ciclo_Guarda_Memoria:
677     MOV R3, [R0] ;Coloca o conteúdo do registo R0 no
        registo R3 (contém o número de proteínas totais na memória principal)
678     MOV R4, [R1] ;Coloca o conteúdo do registo R1 no
        registo R4 (contém o número de proteínas calculadas na memória secundária)
679     ADD R3, R4 ;Adiciona ao valor do registo R3 o valor
        do registo R4 (adiciona o valor das proteínas totais com o valor das proteínas
        calculadas)
680     MOV [R0], R3 ;Coloca na posição de memória R0
        (guardando assim o valor das proteínas na memória principal) o valor do registo R3
681     ADD R0, R2 ;Adiciona ao valor do registo R0 o valor
        do registo R2, assim saltando na memória 10H para localização na memoria
        principal do macronutriente seguinte
682     ADD R1, R2 ;Adiciona ao valor do registo R1 o valor
        do registo R2, assim saltando na memória 10H para localização na memoria
        secundária do macronutriente seguinte
683     SUB R5, 1 ;Subtrai o valor do registo R5 a
        constante 1 (para saber que já percorreu os macronutrientes e calorias)
684     CMP R5, 0 ;Compara o valor do registo R5 com a
        constante 0
685     JNE Ciclo_Guarda_Memoria ;Caso sejam diferentes volta para a tag
        "Ciclo_Guarda_Memoria", quando o contador ficar igual a zero, significa que os
        macronutrientes e as calorias já estão na memoria principal, que será
        apresentada no menu total
686     POP R6
        ;-----
687     POP R5 ;
688     POP R4 ;Retira da pilha os registos guardados
689     POP R3 ;
690     POP R2 ;
691     POP R1 ;
692     POP R0
        ;-----
693     RET
694 ;*****
695 ;Rotina Converte to char
696 ;*****
697
698 Converte_to_CHAR:
699     PUSH R0
700     ;-----
701     PUSH R1 ;
702     PUSH R2 ;
703     PUSH R3 ;Guarda na pilha os registos que são
        alterados na rotina

```



```

703     PUSH R4                                ;
704     PUSH R5                                ;
705     PUSH R8
;-----
706     MOV R0, 10                             ;Coloca em R0 o valor 10 para fazer as
divisões
707     MOV R2, LocalizacaoPesoDisplay         ;Coloca em R2 o endereço base do display
onde ficará visualizado o peso
708     ADD R2, 3                               ;Adiciona ao valor de R2 a constante 3 ,
sendo a osição do 1º caracter preencher
709     MOV R3, 0                               ;Coloca em R3 a constante 0, assim R3
tem número total de caracteres já preenchidos
710
711 ProximoCarater:
712     MOV R4, R8                             ;Coloca em R4 o valor do registo R8,
para não perder o valor ao fazer o calculo do resto
713     MOD R4, R0                             ;R4 fica com o resto da divisao inteira
por R0 (10) -D
714     DIV R8, R0                             ;R1 fica com o quociente da divisão
inteira por R0 (10) -X
715     MOV R5, 48
716     ADD R5, R4                             ;Adiciona 48 ao resto para obter o
caracter em ASCII
717     MOVB [R2], R5                         ;Guarda no display o caracter
718     SUB R2, 1                             ;Subtrai R2 a constante 1 para obter
próxima posição a preencher
719     ADD R3, 1                             ;Adiciona ao registo R3 a constante 1
para incrementar o número total de carateres já preenchidos
720     CMP R8, 0                             ;Verifica se x já é zero
721     JNE ProximoCarater                     ;Caso não seja zero , passa para o
próximo caracter
722
723 PreencherVazios:
724     CMP R3, NR_TOT_CAR                     ;Verifica se já preencheu todos
725     JEQ Fim_Rotina                         ;Se sim termina
726     MOV R5, caraterVazio                  ;Caso contrário, preenche com o carácter
vazio
727     MOVB [R2], R5                         ;Coloca na posição de memória do registo
R2 o valor de R5 (assim preenchendo o resto com caracteres vazios)
728     SUB R2, 1                             ;Próxima posição a preencher
729     ADD R3, 1                             ;Incrementamos o numero total de
carateres ja preenchidos
730     JMP PreencherVazios                   ;Salta para a tag "PreencherVazios"
731
732 Fim_Rotina:
733     POP R8
;-----
734     POP R5                                ;
735     POP R4                                ;
736     POP R3                               ;Retira da pilha os registos guardados
737     POP R2                                ;
738     POP R1                                ;
739     POP R0
;-----
740     RET
741
742 ;*****
743 ;Rotina Converte to char Macronutrientes
744 ;*****
745 Converte_to_CHAR_M:
746     PUSH R0
;-----
747     PUSH R1                                ;
748     PUSH R2                                ;
749     PUSH R3                               ;Guarda na pilha os registos que são
alterados na rotina
750     PUSH R4                                ;
751     PUSH R5                                ;
752     PUSH R6                                ;
753     PUSH R7                                ;
754     PUSH R8
;-----
755     MOV R8, 4                             ;Coloca em R8 a constante 3 (contador)

```

```

756      MOV R7, LocalizacaoMemoriaP          ;Coloca em R7 o endereço da constante
      "LocalizacaoMemoriaP" , esta constante contém a localização na memória onde
      estão guardado as proteínas totais
757      MOV R6,[R7]                          ;Coloca o conteúdo do registo R7 no
      registo R6 (contém o número de proteínas totais)
758      MOV R9, LocalizacaoPDisplay          ;Coloca em R9 o endereço base do display
      onde ficará visualizado as proteínas (para não perder o valor)
759      MOV R2, LocalizacaoPDisplay          ;Coloca em R2 o endereço base do display
      onde ficará visualizado as proteínas
760
761  Ciclo:
762      MOV R0, 10                            ;Coloca em R0 o valor 10 para fazer as
      divisões
763      ADD R2, 3                             ;Adiciona ao valor de R2 a constante 3 ,
      sendo a posição do 1º caracter preencher
764      MOV R3, 0                            ;Coloca em R3 a constante 0, assim R3
      tem número total de caracteres já preenchidos
765
766  ProximoCarater_M:
767      MOV R4, R6                            ;Coloca em R4 o valor do registo R6,
      para não perder o valor ao fazer o calculo do resto
768      MOD R4, R0                            ;R4 fica com o resto da divisao inteira
      por R0 (10) -D
769      DIV R6, R0                            ;R1 fica com o quociente da divisão
      inteira por R0 (10) -X
770      MOV R5, 48
771      ADD R5, R4                            ;Adiciona 48 ao resto para obter o
      caracter em ASCII
772      MOVB [R2], R5                        ;Guarda no display o caracter
773      SUB R2, 1                             ;Subtrai R2 a constante 1 para obter
      próxima posição a preencher
774      ADD R3, 1                             ;Adiciona ao registo R3 a constante 1
      para incrementar o número total de carateres já preenchidos
775      CMP R6, 0                            ;Verifica se x já é zero
776      JNE ProximoCarater_M                ;Caso não seja zero , passa para o
      proximo caracter
777
778  PreencherVazios_M:
779      CMP R3, NR_TOT_CAR                    ;Verifica se já preencheu todos
780      JEQ Fim_Rotina_M                     ;Se sim termina
781      MOV R5, caraterVazio                 ;Caso contrário, preenche com o carácter
      vazio
782      MOVB [R2], R5                        ;Coloca na posição de memória do registo
      R2 o valor de R5 (assim preenchendo o resto com caracteres vazios)
783      SUB R2, 1                             ;Próxima posição a preencher
784      ADD R3, 1                             ;Incrementamos o numero total de
      caracteres ja preenchidos
785      JMP PreencherVazios_M
786
787  Fim_Rotina_M:
788      MOV R1, 10H                          ;Coloca em R1 a constante 10H (16 em
      decimal) para poder passar depois para preencher os seguintes macronutrientes e
      calorias
789      ADD R9, R1                            ;Adiciona o valor R9 o valor R1 para
      passar a linha seguinte
790      MOV R2, R9                            ;Coloca em R2 o valor de R9 (R2 é o
      localização para colocar o próximo macronutriente ou caloria no display)
791      ADD R7, R1                            ;Adiciona o valor R7 o valor R1 (assim
      também na memória salta uma linha acedendo a memória do seguinte macronutriente
      ou calorias)
792      MOV R6, [R7]                          ;Coloca o conteúdo do registo R7 no
      registo R6 (contém o número total do macronutriente seguinte)
793      SUB R8, 1                             ;Subtrai 1 ao contador
794      CMP R8, 0                            ;Compara com o contador com 0
795      JNE Ciclo                            ;Enquanto o contador não é zero vai
      preenchendo e colocando no dislay o valor das proteínas, hidratos de carbono,
      gorduras e calorias no display, caso contrário termina
796      POP R8
      ;-----
797      POP R7                                ;
798      POP R6                                ;
799      POP R5                                ;
800      POP R4                                ;Retira da pilha os registos guardados

```

```

801     POP R3 ;
802     POP R2 ;
803     POP R1 ;
804     POP R0 ;
805     RET
806
807 ;*****
808 ;Rotina Erro
809 ;*****
810 RErro:
811     PUSH R0
812     ;-----
813     PUSH R1 ;Guarda na pilha os registos que são
alterados na rotina
814     MOV R0, OK ;Coloca no registo R0 o endereço do
periférico "OK"
815
816 Erro:
817     MOVB R1, [R0] ;Coloca no registo R1 o valor do carater
do periférico "OK"
818     CMP R1 ,1 ;Compara o registo R1 com o valor
constante 1
819     JNE Erro ;Caso não seja R1 igual a 1 então salta
para o ciclo "Erro"
820     JMP Ligado ;Caso os valores sejam iguais então
salta para tag "Ligado" , significa que o "OK" está ativo então volta para o menu
principal do programa
821     POP R1 ;Retira da pilha os registos guardados
822     POP R0
823     ;-----
824     RET
825
826 ;*****
827 ;Rotina Limpa Display
828 ;*****
829 LimpaDisplay:
830     PUSH R0
831     ;-----
832     PUSH R1 ;Guarda na pilha os registos que são
alterados na rotina
833     PUSH R2
834     ;-----
835     MOV R0, Display_inic ;Coloca em R0 o endereço do início do
display
836     MOV R1, Display_end ;Coloca em R1 o endereço do fim do display
837
838 Ciclolimpa:
839     MOV R2, caraterVazio ;Coloca em R2 o carater vazio
840     MOVB [R0], R2 ;Coloca na memória na posição do R0 o
carater vazio, assim limpando a primeira linha do display
841     ADD R0, 1 ;Incrementa o valor 1 ao valor do
registo R0
842     CMP R0, R1 ;Compara o registo R0 e R1
843     JLE Ciclolimpa ;Verifica se já está no final do
display, caso não tenha chegado ao fim do display salta para "Ciclolimpa"
844     POP R2
845     ;-----
846     POP R1 ;Retira da pilha os registos guardados
847     POP R0
848     ;-----
849     RET
850
851 ;*****
852 ;Rotina Limpa Macronutrientes
853 ;*****
854 LimpaMacronutrientes:
855     PUSH R0
856     ;-----
857     PUSH R1 ;
858     PUSH R2 ;Guarda na pilha os registos que são
alterados na rotina
859     PUSH R3 ;

```

```

853     PUSH R5
854     ;-----
854     MOV R0, LocalizacaoMemoriaP           ;Coloca em R0 o endereço do
"LocalizacaoMemoriaP"
855     MOV R1, LocalizacaoMemoriaC           ;Coloca em R1 o endereço do
"LocalizacaoMemoriaC"
856     MOV R2, LocalizacaoMemoriaF           ;Coloca em R2 o endereço do
"LocalizacaoMemoriaF"
857     MOV R3, LocalizacaoMemoriaCal         ;Coloca em R3 o endereço do
"LocalizacaoMemoriaCal"
858     MOV R5, 0                             ;Coloca em R5 o valor da constante 0
859     MOV [R0], R5                           ;Coloca na posição de memória do
periférico "LocalizacaoMemoriaP" o valor do registo R5, assim limpando a memoria
860     MOV [R1], R5                           ;Coloca na posição de memória do
periférico "LocalizacaoMemoriaC" o valor do registo R5, assim limpando a
memoria
861     MOV [R2], R5                           ;Coloca na posição de memória do
periférico "LocalizacaoMemoriaF" o valor do registo R5, assim limpando a memoria
862     MOV [R3], R5                           ;Coloca na posição de memória do
periférico LocalizacaoMemoriaCal" o valor do registo R5, assim limpando a memoria
863     POP R5
864     ;-----
864     POP R3                                ;
865     POP R2                                ;Retira da pilha os registos guardados
866     POP R1                                ;
867     POP R0
868     ;-----
868     RET
869
870
871 ;*****
872 ;Rotina Limpa Memória Secundária
873 ;*****
874 LimpaMemoriaSecundaria:
875     PUSH R0
876     ;-----
876     PUSH R1                                ;
877     PUSH R2                                ;Guarda na pilha os registos que são
alterados na rotina
878     PUSH R3                                ;
879     PUSH R5
880     ;-----
880     MOV R0, LocalizacaoMemoriaPA           ;Coloca em R0 o endereço do
"LocalizacaoMemoriaPA"
881     MOV R1, LocalizacaoMemoriaCA           ;Coloca em R1 o endereço do
"LocalizacaoMemoriaCA"
882     MOV R2, LocalizacaoMemoriaFA           ;Coloca em R2 o endereço do
"LocalizacaoMemoriaFA"
883     MOV R3, LocalizacaoMemoriaCala         ;Coloca em R3 o endereço do
"LocalizacaoMemoriaCala"
884     MOV R5, 0                             ;Coloca em R5 o valor da constante 0
885     MOV [R0], R5                           ;Coloca na posição de memória do
periférico "LocalizacaoMemoriaPA" o valor do registo R5, assim limpando a
memoria secundária
886     MOV [R1], R5                           ;Coloca na posição de memória do
periférico "LocalizacaoMemoriaCA" o valor do registo R5, assim limpando a
memoria secundária
887     MOV [R2], R5                           ;Coloca na posição de memória do
periférico "LocalizacaoMemoriaFA" o valor do registo R5, assim limpando a
memoria secundária
888     MOV [R3], R5                           ;Coloca na posição de memória do
periférico LocalizacaoMemoriaCala" o valor do registo R5, assim limpando a
memoria secundária
889     POP R5
890     ;-----
890     POP R3                                ;
891     POP R2                                ;Retira da pilha os registos guardados
892     POP R1                                ;
893     POP R0
894     ;-----
894     RET
895
896 ;*****

```

```

897 ;Rotina Limpa Periféricos
898 ;*****
899 LimpaPerifericos:
900     PUSH R0
901     ;-----
902     PUSH R1                                ;
903     PUSH R2                                ;Guarda na pilha os registos que são
904     alterados na rotina
905     PUSH R3                                ;
906     PUSH R4
907     ;-----
908     MOV R0, ON_OFF                        ;Coloca em R0 o endereço do periférico
909     de entrada "ON_OFF"
910     MOV R1, SEL_NR_MENU                    ;Coloca em R1 o endereço do periférico
911     de entrada "SEL_NR_MENU"
912     MOV R2, OK                            ;Coloca em R2 o endereço do periférico
913     de entrada "OK"
914     MOV R3, CHANGE                        ;Coloca em R3 o endereço do periférico
915     de entrada "CHANGE"
916     MOV R4, PESO                          ;Coloca em R4 o endereço do periférico
917     de entrada "PESO"
918     MOV R5, 0                            ;Coloca em R5 o valor da constante 0
919     MOVB [R0], R5                         ;Coloca na posição de memória do
920     periférico "ON_OFF" o valor do registo R5, assim limpando o periférico
921     MOVB [R1], R5                         ;Coloca na posição de memória do
922     periférico "SEL_NR_MENU" o valor do registo R5, assim limpando o periférico
923     MOVB [R2], R5                         ;Coloca na posição de memória do
924     periférico "OK" o valor do registo R5, assim limpando o periférico
925     MOVB [R3], R5                         ;Coloca na posição de memória do
926     periférico "CHANGE" o valor do registo R5, assim limpando o periférico
927     MOV [R4], R5                         ;Coloca na posição de memória do
928     periférico "PESO" o valor do registo R5, assim limpando o periférico
929     POP R4
930     ;-----
931     POP R3                                ;
932     POP R2                                ;Retira da pilha os registos guardados
933     POP R1                                ;
934     POP R0
935     ;-----
936     RET
937
938 ;*****
939 ;Rotina Mostra Display
940 ;*****
941 MostraDisplay:
942     PUSH R0
943     ;-----
944     PUSH R1                                ;Guarda na pilha os registos que são
945     alterados na rotina
946     PUSH R3
947     ;-----
948     MOV R0, Display_inic                  ;Endereço do início do display
949     MOV R1, Display_end                    ;Endereço do final do display
950     ;-----
951     Ciclomostra:
952     MOV R3, [R2]                          ;Copia palavra do menu
953     MOV [R0], R3                          ;Copia coloca no display
954     ADD R2, 2                             ;Incrementa o valor 2 ao valor da
955     palavra do menu
956     ADD R0, 2                             ;Incrementa o valor 2 ao valor da
957     memória do display
958     CMP R0, R1                            ;Compara o valor do registo R0 com o
959     valor do registo R1
960     JLE Ciclomostra                       ;Caso o valor de R0 seja menor ou igual
961     que R1 o ciclo continua, caso seja maior termina
962     POP R3
963     ;-----
964     POP R1                                ;Retira da pilha os registos guardados
965     POP R0
966     ;-----
967     RET

```

```

946 ;*****
947 ;Rotina Desligar
948 ;*****
949 RDesliga:
950     PUSH R0                                ;Guarda na pilha os registos que são
        alterados na rotina
951     PUSH R1
        ;-----
952     MOV R0, ON_OFF                        ;Coloca no registo R0 o endereço do
        periférico "OK"
953     MOVB R1, [R0]                          ;Coloca no registo R1 o valor do carater
        do periférico "OK"
954     CMP R1 ,1                             ;Compara o registo R1 com o valor
        constante 1
955     JNE Fim                               ;Caso seja diferentes então salta paraa
        tag "Fim" e termina a rotina
956     JMP ProgramaPrincipal                ;Caso sejam iguais salta para
        °ProgramaPrincipal" onde o display está vazio
957
958 Fim:
959     POP R1
        ;-----
960     POP R0                                ;Retira da pilha os registos guardados
961     RET
962
963 ;-----
        -----
964 ;Tabela com os macro nutrientes por cada 100 g de um dado alimento
965 ;-----
        -----
966 Place 6000H
967     String " Aveia "
968     String 11,56,7
969 Place 6020H
970     String " Pão de forma "
971     String 9,42,3
972 Place 6040H
973     String " Batata "
974     String 3,19,0
975 Place 6060H
976     String " Arroz "
977     String 7,25,0
978 Place 6080H
979     String " Feijão "
980     String 10,13,0
981 Place 60A0H
982     String " Legumes "
983     String 3,7,0
984 Place 60C0H
985     String " Tomate "
986     String 1,3,0
987 Place 60E0H
988     String " Banana "
989     String 1,23,0
990 Place 6100H
991     String " Laranja "
992     String 1,12,0
993 Place 6120H
994     String " Maçã "
995     String 1,14,0
996 Place 6140H
997     String " Kiwi "
998     String 1,15,0
999 Place 6160H
1000    String " Bolacha Choc "
1001    String 9,59,22
1002 Place 6180H
1003    String " Pizza Q&F PD "
1004    String 13,25,9
1005 Place 61A0H
1006    String " Amêndoas "

```

1007	String	25,6,55	
1008	Place	61C0H	
1009	String	"	Linhaça "
1010	String	18,34,36	
1011	Place	61E0H	
1012	String	"	Azeite "
1013	String	0,0,100	
1014	Place	6200H	
1015	String	"	Leite Magro "
1016	String	3,4,0	
1017	Place	6220H	
1018	String	"	Whey "
1019	String	80,8,4	
1020	Place	6240H	
1021	String	"	Salmão "
1022	String	21,0,15	
1023	Place	6260H	
1024	String	"	Pescada "
1025	String	20,0,1	
1026	Place	6280H	
1027	String	"	Atum "
1028	String	25,0,2	
1029	Place	62A0H	
1030	String	"	Porco "
1031	String	22,0,15	
1032	Place	62C0H	
1033	String	"	Frango (peito) "
1034	String	25,0,4	
1035	Place	62E0H	
1036	String	"	Peru (peito) "
1037	String	28,0,1	
1038	Place	6300H	
1039	String	"	Ovo "
1040	String	7,0,5	
1041	Place	6320H	
1042	String	"	Queijo "
1043	String	28,0,13	