

```

1  /*****
2  Nombre:      p3f3.c
3  Autor:       Programación I
4  Fecha:       08-11-19
5  ACCIONES A REALIZAR:
6
7  Para atender la funcionalidad de la opción 3 del menú se codificará la función
  calcularRangoNotaFinal
8  que asignará una nota alfabética según los siguientes rangos de notas:
9
10 Para almacenar la nota final numérica de forma conjunta con su nota alfabética se
  definirá la
11 estructura tRangoNota:
12
13         typedef struct {
14             double valorNota;
15             char notaAlfabetica;
16         } tRangoNota;
17
18 La función calcularRangoNotaFinal vendrá definida por la siguiente funcionalidad e
  interfaz:
19
20 *****/
21 #include<stdio.h>
22 #include<stdbool.h>
23 #include<windows.h>
24
25 #define NOTAS 4
26 #define NOTA_MINIMA 0.0
27 #define NOTA_MAXIMA 10.0
28
29 typedef struct {
30     double valorNota;
31     char notaAlfabetica;
32 } tRangoNota;
33
34 /*****
35         FASE 2
36 *****/
37
38
39 /*
40 * Función: leerRealEnRango
41 * Lee de la entrada estándar un número real en el rango [rangoInf, rangoSup].
42 * Si se teclea un número fuera de rango lo indica y vuelve a solicitar un nuevo valor.
43 * Parámetros de entrada:
44 *   rangoInf: real, rango inferior del número a leer
45 *   rangoSup: real, rango superior del número a leer
46 * Precondiciones: ninguna
47 * Parámetros de salida pasados por referencia: ninguno
48 * Valor devuelto por la función: número real leído por la entrada estándar (teclado)
49 */
50 double leerRealEnRango (double rangoInf, double rangoSup);
51 double leerRealEnRango(double rangoInf, double rangoSup){
52
53     double num;
54
55     scanf("%lf", &num);
56     fflush(stdin);
57
58     while((num<rangoInf)|| (num>rangoSup)){
59         printf("Valor fuera de rango. Teclee nuevo valor: ");
60         scanf("%lf", &num);
61         fflush(stdin);
62     }
63     return num;
64 }
65
66 /*
67 * Función: leerNotas Parciales
68 * Lee de la entrada estándar las notas de los 4 parciales de la asignatura,
69 * indicando el número de cada parcial (del 1 al 4)
70 * Parámetros de entrada: ninguno
71 * Parámetros de salida pasados por referencia:
72 *   notas: array de reales con las notas de los 4 parciales
73 * Valor devuelto por la función: ninguno
74 */
75 void leerNotasParciales (double notas[]);
76 void leerNotasParciales (double nota[]){
77     int i;
78
79     for(i=0; i<NOTAS; i++){
80         printf("Teclee nota del parcial %d: ", i+1);
81         nota[i]=leerRealEnRango(NOTA_MINIMA, NOTA_MAXIMA);

```

```

82     }
83 }
84 /*
85 * Función: escribirNotasParciales
86 * Escribe en la salida estándar las notas de los 4 parciales de la asignatura,
87 * indicando el número de cada parcial (del 1 al 4)
88 * Parámetros de entrada:
89 * notas: array de reales con las notas de los 4 parciales
90 * Precondiciones: notas tiene que estar inicializado
91 * Parámetros de salida pasados por referencia: ninguno
92 * Valor devuelto por la función: ninguno
93 */
94 void escribirNotasParciales (const double notas[]);
95 void escribirNotasParciales (const double notas[]){
96     int i;
97     printf("\nLas notas obtenidas en los parciales son:\n");
98
99     for(i=0;i<NOTAS;i++){
100         printf("Nota del parcial %d: %g\n",i+1,notas[i]);
101     }
102 }
103 /*
104 * Función: calcularNotaFinal
105 * Calcula la nota final de la asignatura haciendo la media ponderada
106 * de los 4 parciales (15%, 15%, 25% y 45%), respectivamente
107 * Parámetros de entrada:
108 * notas: array de reales con las notas de los 4 parciales
109 * ponderación: array de enteros con la ponderación de cada parcial
110 * Precondiciones: notas y ponderación tienen que estar inicializados
111 * Parámetros de salida pasados por referencia: ninguno
112 * Valor devuelto por la función: valor real con la nota final de la asignatura
113 */
114 double calcularNotaFinal (const double notas[], const int ponderacion[]);
115 double calcularNotaFinal (const double nota[], const int ponderacion[]){
116     int i;
117     double array1[NOTAS], suma=0, notaFinal;
118
119     for(i=0;i<NOTAS;i++){
120         array1[i]=nota[i]*ponderacion[i];
121     }
122     for(i=0;i<NOTAS;i++){
123         suma=suma+array1[i];
124     }
125     notaFinal=(suma/100);
126     return notaFinal;
127 }
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165

```

FASE 3

```

133 /*
134 * Función: modificarNotaParcial
135 * Preguntar si se desea modificar alguna nota y en caso afirmativo se consulta el
136 * número del parcial que se desea modificar y la nueva nota, y se realiza la modificación
137 * Parámetros de entrada:
138 * notas: array de reales con las notas de los 4 parciales
139 * Precondiciones: notas tiene que estar inicializado
140 * Parámetros de salida pasados por referencia:
141 * parcial: número del parcial cuya nota se ha modificado (de 1 a 4)
142 * Valor devuelto por la función: valor booleano, true si se ha modificado alguna nota,
143 * false si no se ha hecho ninguna modificación
144 */
145 bool modificarNotaParcial ( double notas[], int *parcial);
146 /*
147 * Función: leerEnteroEnRango
148 * Lee de la entrada estándar un número entero en el rango [rangoInf, rangoSup].
149 * Si se teclea un número fuera de rango lo indica y vuelve a solicitar un nuevo valor.
150 * Parámetros de entrada:
151 * rangoInf: Mínimo valor entero aceptado como válido
152 * rangoSup: Máximo valor entero aceptado como válido
153 * Precondiciones: Ninguna
154 * Parámetros de salida pasados por referencia: ninguno
155 * Valor devuelto por la función: Valor entero leído por la entrada estándar (teclado)
156 */
157 int LeerEnteroEnRango(int rangoinf,int rangosup);
158 int LeerEnteroEnRango(int rangoinf,int rangosup){
159     int num;
160     scanf("%d",&num);
161     fflush(stdin);
162
163     while((num<rangoinf)|| (num>rangosup)){
164         printf("Valor fuera de rango. Teclee nuevo valor: ");
165         scanf("%d",&num);

```

```

166         fflush(stdin);
167     }
168     return num;
169 }
170
171     /*****
172     FASE 4
173     *****/
174
175 /*
176 * Función: calcularRangoNotaFinal
177 * Calcula la nota alfabética correspondiente a una nota final y almacena ambas en
178 * una estructura del tipo tRangoNota
179 * Parámetros de entrada:
180 * notaFinal: real que contiene la nota final
181 * Precondiciones: notaFinal debe estar calculada previamente
182 * Parámetros de salida pasados por referencia: rangoNota de tipo tRangoNota
183 * Valor devuelto por la función: ninguno
184 */
185 void calcularRangoNotaFinal (double notaFinal, tRangoNota *rangoNota);
186
187 /*
188 * Función: escribirRangoNotaFinal
189 * Presenta por la salida estándar la nota aritmética y la nota alfabética de modo conjunto
190 * Parámetros de entrada:
191 * rangoNota: estructura de tipo tRangoNota
192 * Precondiciones: rangoNota debe de estar inicializado
193 * Parámetros de salida pasados por referencia: ninguno
194 * Valor devuelto por la función: ninguno
195 */
196 void escribirRangoNotaFinal (tRangoNota rangoNota);
197
198 int main(void) {
199     SetConsoleOutputCP(1252);
200     SetConsoleCP(1252);
201
202     double nota[NOTAS], notaFin;
203     int parcial;
204     int i;
205     int opc;
206     leerNotasParciales(nota);
207     escribirNotasParciales(nota);
208     const int porcentajeParcial[4]={15,15,25,45};
209     tRangoNota rangoNota;
210
211
212     do{
213         printf("\n*****Opciones*****\n");
214         printf("\n* 1. Calcular nota final *");
215         printf("\n* 2. Modificar nota parcial *");
216         printf("\n* 3. Calcular rango nota final *");
217         printf("\n* 4. Salir *");
218         printf("\n*****\n");
219         printf("\nTeclee opcion: ");
220         scanf("%d",&opc);
221         fflush(stdin);
222
223         switch(opc){
224             case 1:
225                 printf("Esta usted en la OPCION 1: Calcular nota final.\n");
226                 notaFin = calcularNotaFinal (nota, porcentajeParcial);
227                 printf("La nota final es: %.2f",notaFin);
228
229                 break;
230
231             case 2:
232                 printf("Esta usted en la OPCION 2: Modificar nota parcial.\n");
233                 if(modificarNotaParcial(nota,&parcial)){
234                     printf("Se ha modificado el parcial %d \n",parcial);
235                     escribirNotasParciales(nota);
236                 }
237                 else{
238                     printf("No se ha modificado ninguna nota");
239                 }
240
241                 break;
242
243             case 3:
244                 printf("Esta usted en la OPCION 3:Calcular rango nota final.\n");
245                 notaFin=calcularNotaFinal (nota, porcentajeParcial);
246                 calcularRangoNotaFinal(notaFin,&rangoNota);
247                 escribirRangoNotaFinal(rangoNota);
248
249                 break;

```

```

250
251     case 4:
252         printf("Gracias por utilizar este programa");
253
254         break;
255     default:
256         printf("No es una opcion correcta.Por favor, introduzca una opción valida\n");
257     }
258
259     }while(opc!=4);
260
261     return 0;
262 }
263 /*
264 * Función: modificarNotaParcial
265 * Pregunta si se desea modificar alguna nota y en caso afirmativo se consulta el
266 * número del parcial que se desea modificar y la nueva nota, y se realiza la modificación
267 * Parámetros de entrada:
268 *     notas: array de reales con las notas de los 4 parciales
269 *     Precondiciones: notas tiene que estar inicializada
270 * Parámetros de salida pasados por referencia:
271 *     parcial: número del parcial cuya nota se ha modificado (de 1 a 4)
272 * Valor devuelto por la función: valor booleano, true si se ha modificado alguna nota,
273 *     false si no se ha hecho ninguna modificación
274 */
275 bool modificarNotaParcial(double notas[],int *parcial){
276     bool modificar;
277     char Afirmacion;
278
279     printf(":Desea modificar una nota?[introduce S o s]:\n");
280     scanf("%c",&Afirmacion);
281     fflush(stdin);
282
283     if((Afirmacion=='S')||(Afirmacion=='s')){
284         printf("Teclee el numero de parcial que desea modificar (1,2,3 o 4):");
285         *parcial=LeerEnteroEnRango(1,4);
286
287         printf("Teclee nota del parcial %d: ",*parcial);
288
289         notas[*parcial-1]=leerRealEnRango(NOTA_MINIMA, NOTA_MAXIMA);
290         modificar=true;
291     }
292     else{
293         modificar=false;
294     }
295
296     return modificar;
297 }
298
299 /*
300 * Función: calcularRangoNotaFinal
301 * Calcula la nota alfabética correspondiente a una nota final y almacena ambas en
302 * una estructura del tipo tRangoNota
303 * Parámetros de entrada:
304 *     notaFinal: real que contiene la nota final
305 *     Precondiciones: notaFinal debe estar calculada previamente
306 * Parámetros de salida pasados por referencia: rangoNota de tipo tRangoNota
307 * Valor devuelto por la función: ninguno
308 */
309 void calcularRangoNotaFinal(double notaFinal,tRangoNota *rangoNota){
310
311     rangoNota->valorNota=notaFinal;
312
313     if(notaFinal<5){
314         rangoNota->notaAlfabetica='I';
315     }
316     else
317         if(notaFinal>=5&&notaFinal<7){
318             rangoNota->notaAlfabetica='A';
319         }
320     else
321         if(notaFinal>=7&&notaFinal<9){
322             rangoNota->notaAlfabetica='N';
323         }
324     else
325         if(notaFinal>9){
326             rangoNota->notaAlfabetica='S';
327         }
328 }
329
330 /*****
331 Nombre:    p3f11.c
332 Autor:     Programación I
333 Fecha:     31-10-19

```

```

334     ACCIONES A REALIZAR:
335
336     En el cuerpo de acciones de esta función se debe leer el número del parcial cuya nota se
337     desea modificar.
338     Los números de los parciales son enteros del 1 al 4. Para realizar la lectura del número
339     del parcial se
340     invocará a la función leerEnteroEnRango con la siguiente funcionalidad e interfaz:
341
342     *****/
343     /*
344     * Función: escribirRangoNotaFinal
345     * Presenta por la salida estándar la nota aritmética y la nota alfabética de modo conjunto
346     * Parámetros de entrada:
347     *     rangoNota: estructura de tipo tRangoNota
348     * Precondiciones: rangoNota debe de estar inicializado
349     * Parámetros de salida pasados por referencia: ninguno
350     * Valor devuelto por la función: ninguno
351     */
352     void escribirRangoNotaFinal(tRangoNota rangoNota){
353         printf("La nota final es de %.2f , %c .", rangoNota.valorNota, rangoNota.notaAlfabetica);
354     }

```