

```

1  /*****
2  *****/
3  Nombre:    p3.c
4  Autor:     Programación I
5  Fecha:     21-11-19
6  ACCIONES A REALIZAR:
7
8              //          FASE 1          //
9
10 Se codificará el programa principal para que realice las siguientes acciones:
11
12 - Leer y almacenar las notas de 4 parciales, teniendo en cuenta que las notas son
13 de tipo real
14 y que sólo pueden estar comprendidas entre 0.0 y 10.0.
15
16 - Escribir por la salida estándar las notas leídas previamente, indicando el
17 número del parcial al que
18 corresponden.
19
20 - Escribir el menú que se ha indicado en el aparatado anterior de forma que, si se
21 elige una opción,
22 salga por pantalla una frase indicando que el programa va a ejecutar esa opción y
23 que vuelva a
24 presentar el menú hasta que se elija la opción 4.
25
26              //          FASE 2          //
27
28 Codificar las funciones que se indican a continuación correspondientes a la lectura y
29 escritura de las notas de los 4 parciales e invocarlas desde el programa principal,
30 sustituyendo la parte de código relacionada con estas acciones, codificada en la fase 1.
31
32 Codificar las acciones correspondientes a la opción 1 del menú, Calcular nota final.
33 Para calcular la nota final será necesario declarar en main() una array constante con
34 4 valores de
35 tipo entero que se correspondan con el porcentaje que pondera cada parcial. Estos
36 valores son: 15, 15, 25 y 45.
37
38 Se deberá codificar e invocar la función que se describe a continuación, y una vez
39 obtenido el resultado
40 de la nota final, este se debe escribir por la salida estándar.
41
42              //          FASE 3          //
43
44 En caso de que un alumno haya anotado mal una nota y deba corregirla, o que haya
45 cambiado de nota a
46 consecuencia de una revisión de exámenes, contamos con la opción 2 del menú, Modificar
47 notas parciales.
48
49 Para implementar esta funcionalidad se codificará la función modificarNotaParcial con
50 la siguiente
51 funcionalidad e interfaz:
52
53              //          FASE 4          //
54
55 Para atender la funcionalidad de la opción 3 del menú se codificará la función
56 calcularRangoNotaFinal
57 que asignará una nota alfabética según los siguientes rangos de notas:
58
59 Para almacenar la nota final numérica de forma conjunta con su nota alfabética se
60 definirá la
61 estructura tRangoNota:
62
63     typedef struct {
64         double valorNota;
65         char notaAlfabetica;
66     } tRangoNota;
67
68 La función calcularRangoNotaFinal vendrá definida por la siguiente funcionalidad e
69 interfaz:
70 *****/
71 *****/
72 #include<stdio.h>
73 #include<stdbool.h>          //FASE 3
74 #include<windows.h>          //FASE 4
75
76 #define NOTAS 4              //FASE 1
77 #define NOTA_MINIMA 0.0      //FASE 2
78 #define NOTA_MAXIMA 10.0     //FASE 2
79
80 typedef struct {
81     double valorNota;
82     char notaAlfabetica;

```

```

70 }tRangoNota;
71
72 /*****
73
74 FASE 2
75 *****/
76 /*
77 * Función: leerRealEnRango
78 * Lee de la entrada estándar un número real en el rango [rangoInf, rangoSup].
79 * Si se teclea un número fuera de rango lo indica y vuelve a solicitar un nuevo valor.
80 * Parámetros de entrada:
81 *   rangoInf: real, rango inferior del número a leer
82 *   rangoSup: real, rango superior del número a leer
83 * Precondiciones: ninguna
84 * Parámetros de salida pasados por referencia: ninguno
85 * Valor devuelto por la función: número real leído por la entrada estándar (teclado)
86 */
87 double leerRealEnRango (double rangoInf, double rangoSup);
88
89 /*
90 * Función: leerNotas Parciales
91 * Lee de la entrada estándar las notas de los 4 parciales de la asignatura,
92 * indicando el número de cada parcial (del 1 al 4)
93 * Parámetros de entrada: ninguno
94 * Parámetros de salida pasados por referencia:
95 *   notas: array de reales con las notas de los 4 parciales
96 * Valor devuelto por la función: ninguno
97 */
98 void leerNotasParciales (double notas[]);
99
100 /*
101 * Función: escribirNotasParciales
102 * Escribe en la salida estándar las notas de los 4 parciales de la asignatura,
103 * indicando el número de cada parcial (del 1 al 4)
104 * Parámetros de entrada:
105 *   notas: array de reales con las notas de los 4 parciales
106 * Precondiciones: notas tiene que estar inicializada
107 * Parámetros de salida pasados por referencia: ninguno
108 * Valor devuelto por la función: ninguno
109 */
110 void escribirNotasParciales (const double notas[]);
111
112 /*
113 * Función: calcularNotaFinal
114 * Calcula la nota final de la asignatura haciendo la media ponderada
115 * de los 4 parciales (15%, 15%, 25% y 45%), respectivamente
116 * Parámetros de entrada:
117 *   notas: array de reales con las notas de los 4 parciales
118 *   ponderación: array de enteros con la ponderación de cada parcial
119 * Precondiciones: notas y ponderación tienen que estar inicializados
120 * Parámetros de salida pasados por referencia: ninguno
121 * Valor devuelto por la función: valor real con la nota final de la asignatura
122 */
123 double calcularNotaFinal (const double notas[], const int ponderacion[]);
124
125 /*****
126
127 FASE 3
128 *****/
129 /*
130 * Función: modificarNotaParcial
131 * Pregunta si se desea modificar alguna nota y en caso afirmativo se consulta el
132 * número del parcial que se desea modificar y la nueva nota, y se realiza la modificación
133 * Parámetros de entrada:
134 *   notas: array de reales con las notas de los 4 parciales
135 * Precondiciones: notas tiene que estar inicializado
136 * Parámetros de salida pasados por referencia:
137 *   parcial: número del parcial cuya nota se ha modificado (de 1 a 4)
138 * Valor devuelto por la función: valor booleano, true si se ha modificado alguna nota,
139 *   false si no se ha hecho ninguna modificación
140 */
141 bool modificarNotaParcial ( double notas[], int *parcial);
142
143 /*
144 * Función: leerEnteroEnRango
145 * Lee de la entrada estándar un número entero en el rango [rangoInf, rangoSup].
146 * Si se teclea un número fuera de rango lo indica y vuelve a solicitar un nuevo valor.
147 * Parámetros de entrada:
148 *   rangoInf: Mínimo valor entero aceptado como válido
149 *   rangoSup: Máximo valor entero aceptado como válido
150 * Precondiciones: Ninguna
151 * Parámetros de salida pasados por referencia: ninguno
152 * Valor devuelto por la función: Valor entero leído por la entrada estándar (teclado)
153 */
154 int LeerEnteroEnRango (int rangoinf, int rangosup);

```

```

154 /*****
155
156 FASE 4
157 *****/
158
159 /*
160 * Función: calcularRangoNotaFinal
161 * Calcula la nota alfabética correspondiente a una nota final y almacena ambas en
162 * una estructura del tipo tRangoNota
163 * Parámetros de entrada:
164 * notaFinal: real que contiene la nota final
165 * Precondiciones: notaFinal debe estar calculada previamente
166 * Parámetros de salida pasados por referencia: rangoNota de tipo tRangoNota
167 * Valor devuelto por la función: ninguno
168 */
169 void calcularRangoNotaFinal (double notaFinal, tRangoNota *rangoNota);
170
171 /*
172 * Función: escribirRangoNotaFinal
173 * Presenta por la salida estándar la nota aritmética y la nota alfabética de modo conjunto
174 * Parámetros de entrada:
175 * rangoNota: estructura de tipo tRangoNota
176 * Precondiciones: rangoNota debe de estar inicializado
177 * Parámetros de salida pasados por referencia: ninguno
178 * Valor devuelto por la función: ninguno
179 */
180 void escribirRangoNotaFinal(tRangoNota rangoNota);
181
182 int main(void) {
183
184     SetConsoleOutputCP(1252);
185     SetConsoleCP(1252);
186
187     double nota[NOTAS], notaFin;
188     const int porcentajeParcial[4]={15,15,25,45};
189     int parcial;
190     int opc;
191
192     leerNotasParciales(nota);
193     escribirNotasParciales(nota);
194     tRangoNota rangoNota;
195
196
197     do{
198         printf("\n*****Opciones*****");
199         printf("\n* 1. Calcular nota final *");
200         printf("\n* 2. Modificar nota parcial *");
201         printf("\n* 3. Calcular rango nota final *");
202         printf("\n* 4. Salir *");
203         printf("\n*****\n");
204         printf("\nTeclee opcion: ");
205         scanf("%d",&opc);
206         fflush(stdin);
207
208         switch(opc) {
209
210             case 1:
211                 printf("Esta usted en la OPCION 1: Calcular nota final.\n");
212                 notaFin = calcularNotaFinal (nota, porcentajeParcial);
213                 printf("La nota final es: %.2f",notaFin);
214
215                 break;
216
217             case 2:
218                 printf("Esta usted en la OPCION 2: Modificar nota parcial.\n");
219                 if(modificarNotaParcial(nota,&parcial)){
220                     printf("Se ha modificado el parcial %d \n",parcial);
221                     escribirNotasParciales(nota);
222                 }
223                 else{
224                     printf("No se ha modificado ninguna nota");
225                 }
226
227                 break;
228
229             case 3:
230                 printf("Esta usted en la OPCION 3: Calcular rango nota final.\n");
231                 notaFin = calcularNotaFinal (nota, porcentajeParcial);
232                 calcularRangoNotaFinal(notaFin,&rangoNota);
233                 escribirRangoNotaFinal(rangoNota);
234
235                 break;
236
237             case 4:

```

```

238         printf("Gracias por utilizar este programa.");
239
240         break;
241     default:
242         printf("No es una opcion correcta. Por favor, introduzca una opción valida\n");
243     }
244
245     }while(opc!=4);
246
247     return 0;
248 }
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321

```

```

/*****
FASE 2
*****/

/*
 * Función: leerRealEnRango
 * Lee de la entrada estándar un número real en el rango [rangoInf, rangoSup].
 * Si se teclea un número fuera de rango lo indica y vuelve a solicitar un nuevo valor.
 * Parámetros de entrada:
 *   rangoInf: real, rango inferior del número a leer
 *   rangoSup: real, rango superior del número a leer
 * Precondiciones: ninguna
 * Parámetros de salida pasados por referencia: ninguno
 * Valor devuelto por la función: número real leído por la entrada estándar (teclado)
 */
double leerRealEnRango(double rangoInf, double rangoSup){
    double num;
    scanf("%lf", &num);
    fflush(stdin);

    while((num<rangoInf) || (num>rangoSup)){
        printf("Valor fuera de rango. Teclee nuevo valor: ");
        scanf("%lf", &num);
        fflush(stdin);
    }

    return num;
}

/*
 * Función: leerNotasParciales
 * Lee de la entrada estándar las notas de los 4 parciales de la asignatura,
 * indicando el número de cada parcial (del 1 al 4)
 * Parámetros de entrada: ninguno
 * Parámetros de salida pasados por referencia:
 *   notas: array de reales con las notas de los 4 parciales
 * Valor devuelto por la función: ninguno
 */
void leerNotasParciales (double nota[]){
    int i;

    for(i=0; i<NOTAS; i++){
        printf("Teclee nota del parcial %d: ", i+1);
        nota[i]=leerRealEnRango(NOTA_MINIMA, NOTA_MAXIMA);
    }
}

/*
 * Función: escribirNotasParciales
 * Escribe en la salida estándar las notas de los 4 parciales de la asignatura,
 * indicando el número de cada parcial (del 1 al 4)
 * Parámetros de entrada:
 *   notas: array de reales con las notas de los 4 parciales
 * Precondiciones: notas tiene que estar inicializado
 * Parámetros de salida pasados por referencia: ninguno
 * Valor devuelto por la función: ninguno
 */
void escribirNotasParciales (const double notas[]){
    int i;
    printf("\nLas notas obtenidas en los parciales son:\n");

    for(i=0; i<NOTAS; i++){
        printf("Nota del parcial %d: %g\n", i+1, notas[i]);
    }
}

/*
 * Función: calcularNotaFinal
 * Calcula la nota final de la asignatura haciendo la media ponderada
 * de los 4 parciales (15%, 15%, 25% y 45%), respectivamente
 * Parámetros de entrada:
 *   notas: array de reales con las notas de los 4 parciales
 *   ponderación: array de enteros con la ponderación de cada parcial
 * Precondiciones: notas y ponderación tienen que estar inicializados
 */

```

```

322  * Parámetros de salida pasados por referencia: ninguno
323  * Valor devuelto por la función: valor real con la nota final de la asignatura
324  */
325  double calcularNotaFinal (const double nota[], const int ponderacion[]){
326      int i;
327      double array1[NOTAS], suma=0, notaFinal;
328
329      for(i=0;i<NOTAS;i++){
330          array1[i]=nota[i]*ponderacion[i];
331      }
332      for(i=0;i<NOTAS;i++){
333          suma=suma+array1[i];
334      }
335      notaFinal=(suma/100);
336      return notaFinal;
337  }
338
339      /*****
340          FASE 3
341          *****/
342  /*
343  * Función: modificarNotaParcial
344  * Preguntar si se desea modificar alguna nota y en caso afirmativo se consulta el
345  * número del parcial que se desea modificar y la nueva nota, y se realiza la modificación
346  * Parámetros de entrada:
347  *   notas: array de reales con las notas de los 4 parciales
348  *   Precondiciones: notas tiene que estar inicializado
349  * Parámetros de salida pasados por referencia:
350  *   parcial: número del parcial cuya nota se ha modificado (de 1 a 4)
351  * Valor devuelto por la función: valor booleano, true si se ha modificado alguna nota,
352  *   false si no se ha hecho ninguna modificación
353  */
354  bool modificarNotaParcial(double notas[],int *parcial){
355      bool modificar;
356      char Afirmacion;
357
358      printf("¿Desea modificar una nota?[introduce S o s]:\n");
359      scanf("%c",&Afirmacion);
360      fflush(stdin);
361
362      if((Afirmacion=='S')||(Afirmacion=='s')){
363          printf("Teclee el número de parcial que desea modificar (1,2,3 o 4):");
364          *parcial=LeerEnteroEnRango(1,4);
365
366          printf("Teclee nota del parcial %d: ",*parcial);
367
368          notas[*parcial-1]=leerRealEnRango(NOTA_MINIMA, NOTA_MAXIMA);
369          modificar=true;
370
371      }
372      else{
373          modificar=false;
374
375      }
376      return modificar;
377  }
378
379  /*
380  * Función: leerEnteroEnRango
381  * Lee de la entrada estándar un número entero en el rango [rangoInf, rangoSup].
382  * Si se teclea un número fuera de rango lo indica y vuelve a solicitar un nuevo valor.
383  * Parámetros de entrada:
384  *   rangoInf: Mínimo valor entero aceptado como válido
385  *   rangoSup: Máximo valor entero aceptado como válido
386  * Precondiciones: Ninguna
387  * Parámetros de salida pasados por referencia: ninguno
388  * Valor devuelto por la función: Valor entero leído por la entrada estándar (teclado)
389  */
390  int LeerEnteroEnRango(int rangoinf,int rangosup){
391      int num;
392      scanf("%d",&num);
393      fflush(stdin);
394
395      while((num<rangoinf)|| (num>rangosup)){
396          printf("Valor fuera de rango. Teclee nuevo valor: ");
397          scanf("%d",&num);
398          fflush(stdin);
399      }
400      return num;
401  }
402
403
404      /*****
405          FASE 4

```



```

406          *****/
407
408  /*
409  * Función: calcularRangoNotaFinal
410  * Calcula la nota alfabética correspondiente a una nota final y almacena ambas en
411  * una estructura del tipo tRangoNota
412  * Parámetros de entrada:
413  *   notaFinal: real que contiene la nota final
414  *   Precondiciones: notaFinal debe estar calculada previamente
415  *   Parámetros de salida pasados por referencia: rangoNota de tipo tRangoNota
416  *   Valor devuelto por la función: ninguno
417  */
418  void calcularRangoNotaFinal(double notaFinal, tRangoNota *rangoNota) {
419
420      rangoNota->valorNota = notaFinal;
421
422      if(notaFinal<5){
423          rangoNota->notaAlfabetica = 'I';
424      }
425      else
426          if(notaFinal>=5&&notaFinal<7){
427              rangoNota->notaAlfabetica = 'A';
428          }
429      else
430          if(notaFinal>=7&&notaFinal<9){
431              rangoNota->notaAlfabetica = 'N';
432          }
433      else
434          if(notaFinal>=9){
435              rangoNota->notaAlfabetica = 'S';
436          }
437  }
438
439  /*
440  * Función: escribirRangoNotaFinal
441  * Presenta por la salida estándar la nota aritmética y la nota alfabética de modo conjunto
442  * Parámetros de entrada:
443  *   rangoNota: estructura de tipo tRangoNota
444  *   Precondiciones: rangoNota debe de estar inicializado
445  *   Parámetros de salida pasados por referencia: ninguno
446  *   Valor devuelto por la función: ninguno
447  */
448  void escribirRangoNotaFinal(tRangoNota rangoNota) {
449      printf("La nota final es de %.2f , %c .", rangoNota.valorNota, rangoNota.notaAlfabetica);
450  }
451
452
453
454
455

```