

```

1  /*
2  * Proyecto: IMplementacionDeUnQSortGenerico
3  * Archivo:  FuncionesDeOrdenacionGenericas.h
4  * Autor:    J. Miguel Guanira E. //miguel.guanira.
5  *
6  * Created on 25 de septiembre de 2024, 08:44 AM
7  */
8
9  #ifndef FUNCIONESDEORDENACIONGENERICAS_H
10 #define FUNCIONESDEORDENACIONGENERICAS_H
11
12 void ordenarG(void *arr, int izq, int der, int(*cmp)(const void*,const void *));
13 void cambiarG(void *&arrI,void *&arrK);
14
15 #endif /* FUNCIONESDEORDENACIONGENERICAS_H */
16
17 /*
18 * Proyecto: IMplementacionDeUnQSortGenerico
19 * Archivo:  FuncionesDeOrdenacionGenericas.cpp
20 * Autor:    J. Miguel Gunira E//miguel.guanira.
21 *
22 * Created on 25 de septiembre de 2024, 08:44 AM
23 */
24
25 #include <iostream>
26 #include <iomanip>
27 using namespace std;
28 #include "FuncionesDeOrdenacionGenericas.h"
29
30 void ordenarG(void *arr, int izq, int der, int(*cmp)(const void*,const void *)){
31     void **arreglo = (void**)arr;
32     int limite;
33
34     if(izq>=der) return;
35     cambiarG(arreglo[izq],arreglo[(izq+der)/2]);
36     limite = izq;
37     for (int i = izq+1; i <= der; i++)
38         if(cmp(arreglo[izq],arreglo[i])>0)
39             cambiarG(arreglo[++limite],arreglo[i]);
40     cambiarG(arreglo[izq],arreglo[limite]);
41     ordenarG(arreglo,izq,limite-1,cmp);
42     ordenarG(arreglo,limite+1,der,cmp);
43 }
44
45 void cambiarG(void *&arrI,void *&arrK){
46     void *aux;
47     aux = arrI;
48     arrI = arrK;
49     arrK = aux;
50 }
51
52 /*
53 * Proyecto: IMplementacionDeUnQSortGenerico
54 * Archivo:  main.cpp
55 * Autor:    J. Miguel Guanira E.//miguel.guanira.
56 *
57 * Created on 25 de septiembre de 2024, 08:43 AM
58 */
59
60 #include <iostream>
61 #include <iomanip>
62 using namespace std;
63 #include "FuncionesDeOrdenacionGenericas.h"
64 #include "FunciionesDeEnteros.h"
65 #include "FuncionesDeCadenas.h"
66 #include "FuncionesDePunterosGenericos.h"

```

```

67
68  int main(int argc, char** argv) {
69      //      int a[50]{10,25, 7,15, 8, 33, 45 , 1, 19, 10,6, 16, 41},n=13;
70      //
71      //      ordenarG(a,0,n-1,miCmpInt);
72      //      for (int i = 0; i < n; i++)
73      //          cout<<setw(4)<<a[i];
74      //      cout<<endl;
75
76      //      char **personas;
77      //      int nc;
78      //      leerDatos(personas,nc,"Personas.txt");
79      //      ordenarG(personas,0,nc-1,miStrcmp);
80      //      for (int i = 0; i < nc; i++)
81      //          cout<<personas[i]<<endl;
82
83      void *personal;
84      int np;
85      leerDatos(personal, np,"personal.csv");
86
87      //      ordenarG(personal,0,np-2,miVoidcmpCodigos);
88      //      ordenarG(personal,0,np-2,miVoidcmpNombres);
89      ordenarG(personal,0,np-2,miVoidcmpSueldos);
90      imprimirDatos(personal,np,"ReportePersonal.txt");
91      return 0;
92  }
93
94  /*
95   * Proyecto: UsoDEQsortDeCstdlib
96   * Archivo:  FuncionesDeEnteros.h
97   * Autor:    J. Miguel Guanira E. //miguel.guanira.
98   *
99   * Created on 24 de septiembre de 2024, 09:07 AM
100  */
101
102  #ifndef FUNCIONESDEENTEROS_H
103  #define FUNCIONESDEENTEROS_H
104
105  int miCmpInt(const void *, const void *);
106
107  #endif /* FUNCIONESDEENTEROS_H */
108
109  *
110  * Proyecto: UsoDEQsortDeCstdlib
111  * Archivo:  FuncionesDeEnteros.cpp
112  * Autor:    J. Miguel Gunira E//miguel.guanira.
113  *
114  * Created on 24 de septiembre de 2024, 09:07 AM
115  */
116
117  #include <iostream>
118  #include <iomanip>
119  using namespace std;
120
121  int miCmpInt(const void *a, const void *b){
122      int ai = (int)a, bi= (int)b;
123      return ai-bi;
124  }
125
126  /*
127   * Proyecto: UsoDEQsortDeCstdlib
128   * Archivo:  FuncionesDeCadenas.h
129   * Autor:    J. Miguel Guanira E. //miguel.guanira.
130   *
131   * Created on 24 de septiembre de 2024, 09:13 AM
132   */

```

```

133
134 #ifndef FUNCIONESDECADENAS_H
135 #define FUNCIONESDECADENAS_H
136
137 void leerDatos(char **&personas,int &nc, const char*);
138 char*leeCadena(istream &arch, char delimitador='\n');
139 int miStrcmp(const void*, const void *);
140
141 #endif /* FUNCIONESDECADENAS_H */
142
143 /*
144  * Proyecto: UsoDEQsortDeCstdlib
145  * Archivo:  FuncionesDeCadenas.cpp
146  * Autor:    J. Miguel Gunira E//miguel.guanira.
147  *
148  * Created on 24 de septiembre de 2024, 09:13 AM
149  */
150
151 #include <iostream>
152 #include <fstream>
153 #include <iomanip>
154 using namespace std;
155 #include <cstring>
156 #include "FuncionesDeCadenas.h"
157
158 void leerDatos(char **&personas,int &nc, const char*nombArch){
159     char*buff[200];
160     nc=0;
161     ifstream arch(nombArch,ios::in);
162     if(not arch.is_open()){
163         cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
164         exit(1);
165     }
166     while(true){
167         buff[nc] = leeCadena(arch);
168         if(arch.eof())break;
169         nc++;
170     }
171     personas = new char*[nc];
172     for (int i = 0; i < nc; i++)
173         personas[i] = buff[i];
174 }
175
176 char*leeCadena(istream &arch, char delimitador){
177     char buffer[60], *cad;
178     arch.getline(buffer,60,delimitador);
179     cad = new char[strlen(buffer)+1];
180     strcpy(cad, buffer);
181     return cad;
182 }
183
184 int miStrcmp(const void*a, const void *b){
185     char *ai = (char*)a,*bi = (char*)b;
186     return strcmp(ai,bi);
187 }
188
189 /*
190  * Proyecto: UsoDEQsortDeCstdlib
191  * Archivo:  FuncionesDePunterosGenericos.h
192  * Autor:    J. Miguel Guanira E. //miguel.guanira.
193  *
194  * Created on 24 de septiembre de 2024, 09:34 AM
195  */
196
197 #ifndef FUNCIONESDEPUNTEROSGENERICOS_H
198 #define FUNCIONESDEPUNTEROSGENERICOS_H

```

```

199 void leerDatos(void*&per, int &numDat,const char *nombArch);
200 void *leerRegistro(istream &arch);
201 void aumentamosEspacios(void **&personal,int &numDat,int &cap);
202 void imprimirDatos(void*per, int np, const char *nombArch);
203 void imprimePersona(ofstream &arch,void *per);
204 int miVoidcmpCodigos(const void *a, const void *b);
205 int miVoidcmpNombres(const void *a, const void *b);
206 int miVoidcmpSueldos(const void *a, const void *b);
207
208 #endif /* FUNCIONESDEPUNTEROSGENERICOS_H */
209
210 /*
211  * Proyecto: UsoDEQsortDeCstdlib
212  * Archivo:  FuncionesDePunterosGenericos.cpp
213  * Autor:    J. Miguel Gunira E//miguel.guanira.
214  *
215  * Created on 24 de septiembre de 2024, 09:34 AM
216  */
217
218 #include <iostream>
219 #include <fstream>
220 #include <iomanip>
221 using namespace std;
222 #include <cstring>
223 #include "FuncionesDeCadenas.h"
224 #include "FuncionesDePunterosGenericos.h"
225 #define INCREMENTO 5
226
227 void leerDatos(void*&per, int &numDat,const char *nombArch){
228     ifstream arch(nombArch,ios::in);
229     if(not arch.is_open()){
230         cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
231         exit(1);
232     }
233     void **personal, *reg;
234     int cap=0;
235     numDat =0;
236     personal = nullptr;
237     while(true){
238         reg = leerRegistro(arch);
239         if(arch.eof())break;
240         if(numDat == cap) aumentamosEspacios(personal,numDat,cap);
241         personal[numDat-1] = reg;
242         numDat++;
243     }
244     per = personal;
245 }
246
247 void *leerRegistro(istream &arch){
248     void **registro;
249     int *codigo, cod;
250     char *nombre;
251     double *sueldo;
252     arch >> cod;
253     if(arch.eof()) return nullptr;
254     codigo = new int;
255     *codigo = cod;
256     arch.get();
257     nombre = leeCadena(arch,',');
258     sueldo = new double;
259     arch>>*sueldo;
260     registro = new void*[3]{};
261     registro[0] = codigo;
262     registro[1] = nombre;
263     registro[2] = sueldo;
264     return registro;

```

```

265     }
266
267
268 void aumentamosEspacios(void **&personal,int &numDat,int &cap){
269     void **aux;
270     cap += INCREMENTO;
271     if(personal == nullptr){
272         personal = new void*[cap]{};
273         numDat = 1;
274     }
275     else{
276         aux =new void*[cap]{};
277         for (int i = 0; i < numDat; i++) {
278             aux[i] = personal[i];
279         }
280         delete personal;
281         personal = aux;
282     }
283 }
284
285 void imprimirDatos(void*per, int np, const char *nombArch){
286     ofstream arch(nombArch,ios::out);
287     if(not arch.is_open()){
288         cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
289         exit(1);
290     }
291     void **personal = (void **)per;
292     for (int i = 0; i<np-1 ; i++)
293         imprimePersona(arch,personal[i]);
294 }
295
296 void imprimePersona(ofstream &arch,void *per){
297     void **persona = (void **)per;
298     int *codigo = (int *)persona[0];
299     char*nombre = (char*)persona[1];
300     double *sueldo = (double*)persona[2];
301
302     arch.precision(2);
303     arch<<fixed;
304     arch<<right<<setw(10)<<*codigo<<" " <<left<<setw(40)<<nombre<<right
305         <<setw(10)<<*sueldo <<endl;
306
307 }
308
309 int miVoidcmpCodigos(const void *a, const void *b){
310     void **regA = (void **)(a), **regB = (void**)(b);
311     int *codA = (int*)regA[0], *codB=(int*)regB[0];
312     return *codA-*codB;
313 }
314
315 int miVoidcmpSueldos(const void *a, const void *b){
316     void **regA = (void **)(a), **regB = (void**)(b);
317     double *codA = (double*)regA[2], *codB=(double*)regB[2];
318     return (int) (*codA-*codB);
319 }
320
321 int miVoidcmpNombres(const void *a, const void *b){
322     void **regA = (void **)(a), **regB = (void**)(b);
323     char *nombA = (char*)regA[1], *nombB=(char*)regB[1];
324     return strcmp(nombA,nombB);
325 }

```