



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en computación

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Kevin Alejandro Gonzalez Torres

Matrícula: 372354

Maestro: Pedro Núñez Yépiz

Actividad Numero: 10

Tema - Unidad: Estructuras

Ensenada Baja California a 15 de octubre del 2022



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

En el lenguaje de programación C, las estructuras son una característica fundamental que permite a los programadores definir tipos de datos personalizados para organizar y almacenar múltiples variables relacionadas bajo un solo nombre. Estas estructuras son una forma eficiente de representar objetos complejos en un programa, ya que permiten agrupar datos de diferentes tipos en una sola entidad.

En esencia, una estructura en C es una colección de variables de diferentes tipos de datos (como enteros, flotantes, caracteres, punteros, etc.) que se agrupan bajo un nombre común. Cada variable en una estructura se denomina "miembro" o "campo". Estos campos pueden tener nombres significativos que facilitan la manipulación de los datos.

La palabra clave typedef crea un alias "Punto" para la estructura "struct Punto", lo que facilita la declaración y uso de variables de tipo "Punto" en el código.

En resumen, las estructuras en C son una forma esencial de organizar y manipular datos complejos, y typedef es una característica útil para simplificar la declaración y uso de estructuras en el código.

2. COMPETENCIA

Se practicará y se aprenderá el uso de las estructuras y la definición de typedef en C. Así como también se reforzarán aprendizajes de clases anteriores como lo son los ciclos o las estructuras de control.

3. FUNDAMENTOS

Typedef:

<https://www.geeksforgeeks.org/typedef-in-c/>

<https://learn.microsoft.com/es-es/cpp/c-language/typedef-declarations?view=msvc-170>

Struct

https://www.w3schools.com/c/c_structs.php

<https://www.geeksforgeeks.org/structures-c/>



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

4. PROCEDIMIENTO

REALICE EL SIGUIENTE PROGRAMA QUE CONTenga UN MENÚ.

MENÚ

- 1.- AGREGAR (AUTOM 10 REGISTROS)
- 2.- AGREGAR MANUAL
- 3.- ELIMINAR REGISTRO (lógico)
- 4.- BUSCAR
- 5.- ORDENAR
- 6.- IMPRIMIR
- 0.- SALIR

UTILIZAR UN ARREGLO DE 500 REGISTROS

SE DEBERÁ **UTILIZAR ESTRUCTURAS** CON LOS DATOS BÁSICOS DE UN ALUMNO (status, Matricula, ApPat, ApMat, Nombre, Edad, Sexo)

Busqueda y Ordenacion por campo MATRICULA



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5. RESULTADOS Y CONCLUSIONES



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1.-

```
65 case 1:
66     if (i + 1 > 500)
67     {
68         printf("Register full\n");
69     }
70     sorted = 0; // Verify if it is sorted or not
71
72     for (int j = 0; j < 500; j++)
73     {
74         if (j + 1 > 500)
75         {
76             printf("Register full\n");
77         }
78         temp = autoDataReg();
79         while (linearSearch(studentArray, i, temp.enrolment) != -1)
80         {
81             temp.enrolment = randomNumber(300000, 399999);
82         }
83         studentArray[i++] = temp;
84     }
85
86     if (i + 1 > 500)
87     {
88         printf("Register full\n");
89     }
90
91     break;
```

```
187 Tstudents autoDataReg()
188 {
189     Tstudents student;
190     student.status = 1;
191
192     student.enrolment = randomNumber(300000, 399999);
193
194     strcpy(student.fatherLastname, lastnames[randomNumber(0, 19)]);
195     strcpy(student.motherLastname, lastnames[randomNumber(0, 19)]);
196
197     if (randomNumber(0, 1) == 1)
198     {
199         strcpy(student.name, maleNames[randomNumber(0, 19)]);
200         student.gender = 1;
201     }
202     else
203     {
204         strcpy(student.name, femaleNames[randomNumber(0, 19)]);
205         student.gender = 0;
206     }
207
208     student.age = randomNumber(17, 30);
209
210     return student;
211 }
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

2.-

```
93         case 2:
94
95             sorted = 0;
96
97             temp = manualDataReg();
98
99             while (linearSearch(studentArray, i, temp.enrolment) != -1)
100             {
101                 printf("Repeated enrolment - Please enter a different one.");
102                 temp.enrolment = validate(300000, 399999);
103             }
104             studentArray[i++] = temp;
105
106             if (i + 1 > 500)
107             {
108                 printf("Register full\n");
109             }
110
111             break;
```

```
213 Tstudents manualDataReg()
214 {
215     Tstudents student;
216
217     student.status = 1;
218
219     printf("\nEnrolment:\n");
220     student.enrolment = validate(300000, 399999);
221
222     printf("\nFather Lastname:\n");
223     validateString(student.fatherLastname, 30);
224
225     printf("\nMother Lastname:\n");
226     validateString(student.motherLastname, 30);
227
228     printf("\nName:\n");
229     validateString(student.name, 30);
230
231     printf("\nAge:\n");
232     student.age = validate(17, 30);
233
234     printf("\nGender:\n");
235     student.gender = validate(0, 1);
236     return student;
237 }
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

3.-

```
113 case 3:
114     printf("Enrolment to be removed: ");
115     remove = validate(300000, 399999);
116     search = linearSearch(studentArray, i, remove);
117
118     if (linearSearch(studentArray, i, remove) != -1)
119     {
120         printf("Enrolment found...\n");
121     }
122     else
123     {
124         printf("Enrolment not found...\n");
125     }
126
127     if (studentArray[search].status == 0)
128     {
129         printf("Enrolment already removed.");
130     }
131     else
132     {
133         studentArray[search].status = 0;
134         printf("Now the enrolment %i has been removed.", remove);
135     }
136     break;
```

4.-

```
138 case 4:
139     if (sorted == 0)
140     {
141         sorted = sortVector(studentArray, i);
142         printf("Register is now sorted.");
143     }
144     else
145     {
146         printf("Register already sorted.");
147     }
148     break;
149
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5.-

```
150     case 5:
151         printf("Search enrolment: ");
152         search = validate(300000, 399999);
153
154         if (sorted == 0)
155         {
156             found = linearSearch(studentArray, i, search);
157         }
158         else
159         {
160             found = binarySearch(studentArray, 300000, 399999, search);
161         }
162
163         if (found != -1)
164         {
165             printf("Enrolment found in register number %i.", found);
166         }
167         else
168         {
169             printf("Enrolment not found.");
170         }
171         break;
239 int linearSearch(Tstudents array[], int size, int searchNumber)
240 {
241     int i;
242
243     for (i = 0; i < size; i++)
244     {
245         if (array[i].enrolment == searchNumber)
246         {
247             return i;
248         }
249     }
250     return -1;
251 }
252
253 int binarySearch(Tstudents array[], int left, int right, int number)
254 {
255     while (left <= right)
256     {
257         int medium = left + (right - left) / 2;
258
259         if (array[medium].enrolment == number)
260             return medium;
261
262         if (array[medium].enrolment < number)
263             left = medium + 1;
264
265         else
266             right = medium - 1;
267     }
268
269     return -1;
270 }
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

6.-

```
173 case 6:
174     printRegister(studentArray, i);
175     break;
176
```

```
291 void printRegister(Tstudents array[], int size)
292 {
293     int i;
294
295     printf("STATUS  ENROLMENT  NAME                      FATHER LN          MOTHER LN          AGE  GENDER  \n\n")
296     for (i = 0; i < size; i++)
297     {
298         if (array[i].status == 1)
299         {
300             printf("%-3d  %-9d  %-30s  %-30s  %-30s  %-4d  %-3d\n",
301                 array[i].status, array[i].enrolment, array[i].name, array[i].fatherLastname, array[i].motherLastname, array[i].age, array[i].gender);
302             }
303         }
304     }
305
```

6. ANEXOS

GTKA_RP10_PE_ANEXOS



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

7. REFERENCIAS

Diseño de algoritmos y su codificación en lenguaje C

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

Programación estructurada a fondo: implementación de algoritmos en C

:Pearson Educación. Sznajdleder, P. A. (2017)..

Buenos Aires, Argentina: Alfaomega

Como programar en C/C++

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN: 9688804711

Programación en C. Metodología, estructura de datos y objetos

Joyanes, L. y Zahonero, I. (2001)..

España: McGraw-Hill.

ISBN: 8448130138