

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Estructura de Datos y Algoritmos I

Actividad de lunes no. 1: Repaso post-paro.

Torres Oropeza Diego Alberto

07/06/2021



Los documentos y cosas que utilicé para repasar lo visto en FP y el tema de arreglos fueron las primeras 2 actividades del curso, en particular las siguientes partes:

Condicionales en Lenguaje C

Como en Pseint, en Lenguaje C hay estructura condicionales, esta vez es el “if” y para opciones el “switch”.

- “if”: Es la estructura de “Si” en C, tiene los siguientes componentes:

```
if (n==0)
{
}
else
```

Dentro de los corchetes puedes poner una serie de acciones si se cumple la condición o condiciones que declaras en el paréntesis de enseguida del “if”, si no se cumplen las condiciones y quieres dar otra serie de acciones, se pone el “else” que sería como el “SiNo” de Pseint.

- “switch”: Estructura de selección en C, tiene la siguiente forma:

```
switch(opc)
1:
break;
2:
break;
default:
```

Lo de dentro del paréntesis es la variable de la cual dependerá la opción, entre cada número y cada break puedes poner una serie de instrucciones, el break sirve para indicar que hasta ahí llega la acción, si es algo muy grande lo que se hace, se pueden usar las llaves para no tener muchas complicaciones o funciones que veremos más adelante. El “default” actúa como un “SiNo”, sólo que no es exacto obviamente, por ejemplo, tienes opciones del 1 al 5 y el usuario mete un 6, el default sirve para que en casos como este, en el que el usuario introduce datos no permitidos, el código le puede notificar que está mal, es cuestión de decidir si se quiere presentar en el ejecutable un printf(“Opcion no permitida”);, o si simplemente quieres dejar que el programa se cierre.

Estructuras cíclicas en Lenguaje C

Las estructuras cíclicas que vimos fueron 3: “while”, “for” y “do-while”, las 3 con el mismo propósito de

repetir una serie de acciones mientras su condición se cumpla, pero con diferentes formas cada una.

- “for”: Se declara un valor inicial del contador, el límite del valor del contador y el incremento o decremento según sea el caso, la estructura es la siguiente:

```
for(i=0;i<=10;i++)  
{  
}
```

Dentro de las llaves van las instrucciones a hacer, “i” es el contador, se debe declarar a inicios del programa, los “++” indican que después de que el ciclo realice una vuelta, se le incrementará 1 a “i”, así hasta que “i” llegue a 10 en este caso, aunque el límite no solo puede ser un número, si no también una variable.

- “while”: Aquí sólo se declara el límite del contador y dentro del ciclo se le debe dar el incremento:

```
i=0;  
while(i<=10)  
{  
    i++;  
}
```

- “do-while”: Es parecido al “while”, en el nombre lo dice, pero en este primero se hace la secuencia de acciones y se suma el contador y finalmente comprueba que no se pase del límite:

```
i=0;  
do  
{  
    i++;  
}  
while(i<=10);
```

También repasé con algunos códigos que ya tenía hechos:

Matrices:

```
Start here  X  Ciclo mientras en C.c  X  Matriz transpuesta.c  X

1  #include <stdio.h>
2  #include <conio.h>
3  #define pf printf
4  #define sf scanf
5  #define max 10
6
7  int main(){
8      int matriz[max][max], col, reng, i=0, j=0;
9
10     system("color 3");
11     pf("No. de columnas:"); //Leer cant. de datos
12     sf("%d", &col);
13     pf("No. de renglones:");
14     sf("%d", &reng);
15
16     for(i=0; i<reng; i++) //Leer coord. de Matriz
17     {
18         for(j=0; j<col; j++)
19         {
20             pf("Valor de coordenada [%d][%d]-\t", i+1, j+1);
21             sf("%d", &matriz[i][j]);
22         }
23     } //FIN LEER DATOS
24
25     pf("\nMatriz original:\n"); //matriz original
26     for(i=0; i<reng; i++)
27     {
28         pf("\n\t\t");
29         for(j=0; j<col; j++)
30         {
31             pf("%6d", matriz[i][j]);
32         }
33     }
34
35     pf("\nMatriz transpuesta:\n"); //matriz transpuesta
36     for(i=0; i<col; i++)
37     {
38         pf("\n\t\t");
39         for(j=0; j<reng; j++)
40         {
41             pf("%6d", matriz[j][i]);
42         }
43     }
44 }
45

C:\Users\Bienvenido\Documents\FP\Matriz transpuesta.c
```

Funciones:

```
Start here X Cido mientras en C.c X Matriz transp
3  #define pf printf
4  #define sf scanf
5
6  float fpideValor()
7  {
8      float nl;
9      pf("Dame valor:\n");
10     sf("%f",&nl);
11     return nl;
12 }
13
14 float FunSuma(float a,float b)
15 {
16     float r;
17     r=(a+b);
18     return r;
19 }
20
21 void FunImprimeResultado(float r)
22 {
23     pf("Resultado = %.2f\n",r);
24 }
25
26 int main() {
27
28     float a,b,r;
29     system("color 5");
30     a=fpideValor();
31     b=fpideValor();
32     r=FunSuma(a,b);
33     FunImprimeResultado(r);
34 }
35
```

C:\Users\Bienvenido\Documents\FP\suma funciones.c

Estructuras de datos:

Estructuras de datos, n estudiantes.c - Code::Blocks 13.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plug

Start here x Ejercicio 1 apuntes en clase.c x Estructuras de datos ejemplo clase 2.c

```
1  #include <stdio.h>
2  #include <conio.h>
3  #define pf printf
4  #define sf scanf
5  int i,n;
6  struct datosAlumnos
7  {
8      char nombre[30];
9      char apellido[40];
10     char carrera[40];
11     char numCta[10];
12     int edad;
13     float estatura;
14     float peso;
15 };
16
17 struct datosAlumnos estudiante[3];
18
19 int main()
20 {
21     pf("%cCuántos alumnos?\n",168,160);
22     sf("%i",&n);
23     for(i=0;i<n;i++)
24     {
25         fflush(stdin);
26         pf("Escribe el nombre del alumno\n");
27         gets(estudiante[i].nombre);
28         fflush(stdin);
29         pf("Escribe el apellido del alumno\n");
30         gets(estudiante[i].apellido);
31         fflush(stdin);
32         pf("Escribe la carrera del alumno\n");
33         gets(estudiante[i].carrera);
34         fflush(stdin);
35         pf("Escribe el número de cuenta del alumno\n",163);
36         gets(estudiante[i].numCta);
37         fflush(stdin);
38         pf("Escribe la edad del alumno\n");
39         sf("%i",&estudiante[i].edad);
40         pf("Escribe la estatura del alumno\n");
41         sf("%f",&estudiante[i].estatura);
42         pf("Escribe el peso del alumno\n");
43         sf("%f",&estudiante[i].peso);
44     }
45     for(i=0;i<n;i++)
46     {
47         pf("Los datos del alumno %s %s con no. de cuenta %s son: \n",estudiante[i].nombre,estudiante[i].apellido,estudiante[i].numCta);
48         pf("Carrera: %s\n",estudiante[i].carrera);
49         pf("Edad: %i \n",estudiante[i].edad);
50         pf("Estatura: %f \n",estudiante[i].estatura);
51         pf("Peso: %f\n",estudiante[i].peso);
52     }
53 }
54
```

Para repasar el tema de arreglos vi un video en YouTube que me pareció súper completo, dejo el link en las referencias. También repasé los códigos para realizar la escítala espartana, el cifrado César y el sudoku.

Referencias:

- Tista, E. (2020, 28 septiembre). Todo lo que hay saber sobre Arreglos [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=a4mdXRMq1VE&feature=youtu.be>