UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE INGENIERÍA

ESTRUCTURA DE DATOS Y ALGORITMOS I

Actividad 1: Repaso lenguaje C

Laura Mildred Moreno Razo

FECHA: 07/06/2021

Es un lenguaje de propósito general basado en el paradigma estructurado. El teorema del programa estructurado dicta que todo programa puede desarrollarse utilizando únicamente 3 instrucciones de control:

Secuencia

Selección

Iteración

Consiste en una o más funciones, de las cuales una de ellas debe llamarse main( ) y es la principal.

Comentarios. Una línea "//" Varias líneas "/*" "*/"

Agregar bibliotecas. #include Entrada y salida estándar. stdio.h

Otras stdlib.h math.h string.h

Declarar variables. [modificadores] tipoDeDato identificador [= valor];

Tipos de variable

Enteras

Tipo	*Bits*	*Valor Mínimo*	*Valor Máximo*
signed char	8	-128	127
unsigned char	8	0	255
signed short	16	-32 768	32 767
unsigned short	16	0	65 535
signed int	32	-2,147,483,648	2 147 483 647
unsigned int	32	0	4 294 967 295
signed long	64	9 223 372 036 854 775 808	9 223 372 036 854 775 807
unsigned long	64	0	18 446 744 073 709 551 615
enum	16	-32 768	32 767

Si se omite el clasificador por defecto se considera 'signed'.

Reales

Tipo	*Bits*	*Valor Mínimo*	*Valor Máximo*
float	32	3.4 E-38	3.4 E38
double	64	1.7 E-308	1.7 E308
long double	80	3.4 E-4932	3.4 E4932

Tipo de dato	Especificador formato
Entero	%d Entero en base 10 %i
	%ld Entero largo en base 10 %li
	%o Entero base 8
	%x Entero base 16
Flotante	%f reales de precisión simple
	%lf a reales de doble precisión
	%e notación científica
	%g (redondea la parte fraccionaria a 3 dígitos significativos
Carácter	%c carácter
	%d valor del código ASCII del carácter en base 10 %i
	%o valor del código ASCII del carácter en base 8
	%x valor del código ASCII del carácter en base 16
Cadena de caracteres	%s (Arreglo)

Identificador. Notación de camello, los nombres de cada palabra empiezan con mayúscula y el resto se escribe con minúsculas

printf es una función para imprimir con formato

printf("El valor de la variable real es: %lf", varReal);

scanf es una función que sirve para leer datos de la entrada estándar (teclado)

scanf ("%i", &varEntera);

Secuencias escape.

\a carácter de alarma

\b retroceso

\f avance de hoja

\n salto de línea

\r regreso de carro

\t tabulador horizontal

\v tabulador vertical

'\0' carácter nulo

Modificadores. Van al inicio de la declaración de variables son const (impide que una variable cambie su valor durante la ejecución del programa) y static (indica que la variable permanece en memoria desde su creación y durante toda la ejecución del programa, es decir, permanece estática en la memoria)

Operadores

Aritmético

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo

Lógico

Operador	*Operación*
>>	Corrimiento a la derecha
<<	Corrimiento a la izquierda
&	Operador AND
|	Operador OR
~	Complemento ar-1

Operador	*Operación*
==	Igual que
!=	Diferente a
<	Menor que
>	Mayor que
< =	Menor o igual
> =	Mayor o igual

Operador	*Operación*
!	No
&&	Y
||	O

Estructuras de selección

If

```
if (expresión_lógica) {
// bloque de código a ejecutar
}
```

If-else

```
if (expresión_lógica) {
 // bloque de código a ejecutar
```

```
 // si la condición es verdadera
} else {
 // bloque de código a ejecutar
 // si la condición es falsa
}
```

Switch

```
switch (opcion_a_evaluar){
 case valor1:
 /* Código a ejecutar*/
 break;
 case valor2:
 /* Código a ejecutar*/
 break;
 …
 case valorN:
 /* Código a ejecutar*/
 break;
 default:
 /* Código a ejecutar*/
}
```

Operador ternario. Condición ? SiSeCumple : SiNoSeCumple

Estructuras de repeticion

While

```
while (expresión_lógica) {
 // Bloque de código a repetir
 // mientras que la expresión
```

```
// lógica sea verdadera.
}
```

Do-while

```
do {
 /*
 Bloque de código que se ejecuta
 por lo menos una vez y se repite
 mientras la expresión lógica sea
 verdadera.
 */
} while (expresión_lógica);
```

For

```
for (inicialización  ; expresión_lógica ; operaciones por iteración) {
 /*
 Bloque de código
 a ejecutar
 */
}
```

Arreglos. tipoDeDato nombre[tamaño]

Multidimensionales. tipoDato nombre[tamaño][tamaño]…[tamaño];

Declarar apuntadores. TipoDeDato *apuntador, variable; apuntador = &variable;

Cuando a una variable le antecede un ampersand, se hace es accede a la dirección de memoria de la misma

Funciones

valorRetorno nombre (parámetros){ // bloque de código de la función }

Firma función valorRetorno nombre (parámetros);

Abrir archivo *FILE fopen(char *nombre_archivo, char *modo

Cerrar archivo int fclose(FILE *apArch);

Modos

r: Abre un archivo de texto para lectura.

w: Crea un archivo de texto para escritura.

a: Abre un archivo de texto para añadir.

r+: Abre un archivo de texto para lectura / escritura.

w+: Crea un archivo de texto para lectura / escritura.

a+: Añade o crea un archivo de texto para lectura / escritura.

rb: Abre un archivo en modo lectura y binario.

wb: Crea un archivo en modo escritura y binario.

Leer y escribir archivos

char *fgets (char *buffer, int tamaño, FILE *apArch);

char *fputs (char *buffer, FILE *apArch);

int fprintf (FILE *apArch, char *formato, ...);

int fscanf (FILE *apArch, char *formato, ...);

Un arreglo es un conjunto de datos del mismo tipo (enteros, flotantes, caracteres, etc.) que se almacenan de manera consecutiva y con una posición especifica en la memoria. Para acceder a este espacio se utilizan índices dependiendo de la manera en que estén organizados.

Arreglos unidimensionales. La colección de datos esta ordenada en una sola dimensión, es decir podemos imaginarla como una fila o una columna con n elementos.

Arreglos multidimensionales. Poseen dos o más dimensiones, dentro de las cuales se guardan los datos.