Universidad Andrés Bello

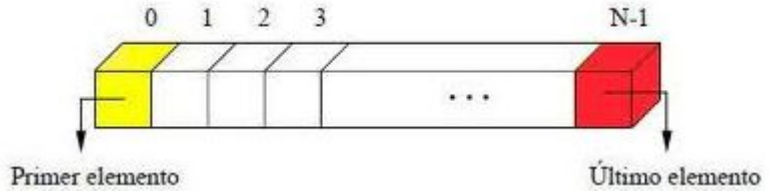
Lenguajes de programación - Clase 2
Arreglos

Arreglos

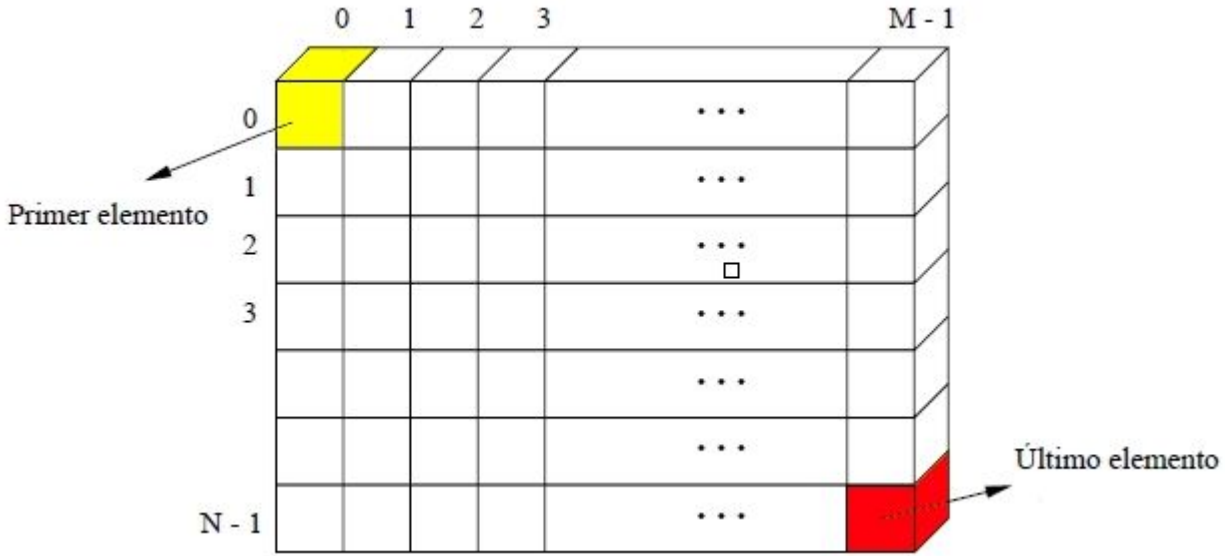
Los arreglos son una estructura de datos de longitud finita, en el cual se almacenan elementos (de un mismo tipo de dato) de forma consecutiva en memoria.

Arreglos

- *Unidimensionales*



- *Multidimensionales*



Arreglos en C

Para declarar un arreglo en C debemos realizar lo siguiente:

Arreglos en C

Universidad Andrés Bello

Ejemplos:

```
int mi_arreglo_mult [33][8];

char mi_string [10];

mi_string='c';

int mi_arreglo[] = {0,1,1,2,3,5,8,13};

int mi_arreglo[2][4] = {{0,1,1,2},{3,5,8,13}};

int mi_arreglo[2][4] ={{ 0,1,1,2},
                       {3,5,8,13}};
```

Arreglos

Recorrer un arreglo en C

```c
#include <stdio.h>

int main(){

	int mi_arreglo[] = {0,1,1,2,3,5,8,13};
	int i;
	int tamanio_arreglo = (sizeof(mi_arreglo) / sizeof(mi_arreglo[0]));
	for (i = 0; i < tamanio_arreglo; i++)
	{
		printf("mi_arreglo[%i] = %i \n ", i, mi_arreglo[i]);

	}

	return 0;
}
```

```
mi_arreglo[0] = 0
mi_arreglo[1] = 1
mi_arreglo[2] = 1
mi_arreglo[3] = 2
mi_arreglo[4] = 3
mi_arreglo[5] = 5
mi_arreglo[6] = 8
mi_arreglo[7] = 13

------------------
(program exited with code: 0)
Presione una tecla para continuar . . .
```

Arreglos

Recorrer un arreglo bidimensional

```c
#include <stdio.h>

int main(){

	int mi_arreglo[2][4] = {{0,1,1,2},{3,5,8,13}};

	int i,j;

	for (i = 0; i < 2; i++)
	{
		for (j = 0; j < 4; j++)
		{
			printf("arreglo en la posición[%i][%i]=%i \n",i,j,mi_arreglo[i][j]);
		}
	}
```

```
arreglo en la posicion[0][0]=0
arreglo en la posicion[0][1]=1
arreglo en la posicion[0][2]=1
arreglo en la posicion[0][3]=2
arreglo en la posicion[1][0]=3
arreglo en la posicion[1][1]=5
arreglo en la posicion[1][2]=8
arreglo en la posicion[1][3]=13
```

Arreglos

Arreglos y funciones (ejemplo)

```c
#include <stdio.h>
#include <time.h>

void mostrar(int arreglo[],int n);
int* generar(int n);

int main(){

	int n=5;
	int i,random;
	int* arreglo = generar(n);
	mostrar(arreglo,n);
	/*for (i = 0; i < n; i++)
	{
		printf("%d \n",*arreglo++);
	}*/
	//generar(n);

	return 0;
}
```

```c
int* generar(int n){

	static int arreglo[100];
	int i, random;
	srand (time(NULL));
	for (i = 0; i < n; i++)
	{
		random = rand() % 33;
		arreglo[i]=random;
	}

	return arreglo;
}

void mostrar(int arreglo[], int n ){
	int i;
	for (i = 0; i < n; i++)
	{
		printf("%d \n",arreglo[i]);
	}
}
```

```
4
9
2
15
24


------------------
(program exited with code: 0)

Presione una tecla para continuar . . .
```

Arreglos

Arreglo de caracteres

```
char mi_cadena[] = "Hola";

char mi_cadena[] = {'H','o','l','a',0};

char mi_cadena[] = {'H','o','l','a','\0'};

char mi_cadena[5] = "Hola";

char mi_cadena[5] = {'H','o','l','a',0};

char mi_cadena[5] = {'H','o','l','a','\0'};
```

***Los arreglos de caracteres siempre terminan con el carácter especial '\0'**
****Debido a este carácter, cuando declaramos nuestro arreglo debemos considerarlo en el tamaño de éste (es decir el tamaño debe ser n+1).***

Arreglos

Arreglos de caracteres (ejemplo)

```c
#include <stdio.h>

int main(){

	char mi_string[]="Hola !!";
	int n = (sizeof(mi_string)/sizeof(mi_string[0]));

	printf("Las letras de mi cadena son: \n");
	for (int i = 0; i < n; i++)
	{
		printf("%c \n", mi_string[i]);
	}
	printf("==========================\n");
	printf("y \n\n");

	printf("Mi cadena en ascii es: \n");
	for (int i = 0; i < n; i++)
	{
		printf("%i \n", mi_string[i]);
	}

	return 0;
}
```

```
Las letras de mi cadena son:
H
o
l
a

!
!

==========================
y

Mi cadena en ascii es:
72
111
108
97
32
33
33
0


------------------
(program exited with code: 0)

Presione una tecla para continuar . . .
```

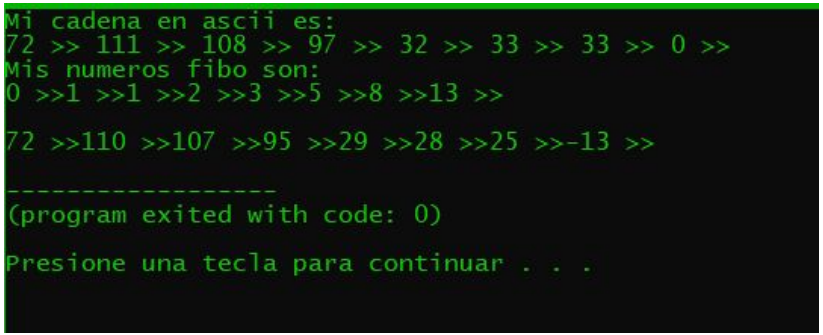
Universidad Andrés Bello

Ejercicio 2:

Resta de arreglos

Instrucciones:

1. ***Genere un arreglo de caracteres con la cadena que desee.***
2. ***Genere un arreglo de enteros cuyo largo debe ser el mismo que el de su arreglo de caracteres.***
3. ***El arreglo de enteros debe ser llenado con los números fibonacci desde 0 hasta el largo de su arreglo.***
4. ***Puede implementar la función Fibonacci de forma recursiva o iterativa.***
5. ***Muestre por pantalla el código ASCII correspondiente a su cadena de caracteres.***
6. ***Muestre por pantalla la secuencia Fibonacci obtenida.***
7. ***Realice la resta entre el ASCII de su cadena de caracteres y la secuencia Fibonacci.***
8. ***Muestre los números resultantes por pantalla.***



Universidad Andrés Bello

No olvidar!

repl.it