**VECTORES Y APUNTADORES - EJERCICIOS**

****

**Equipo de Trabajo**

Yaqueline Hoyos Montes

Juan Pablo Tobón

**UNIVERSIDAD DE ANTIOQUIA**

**MEDELLÍN**

**2018**

## 

## **1. Ejercicios conceptuales**

**Ejercicio 1 - Análisis de código**:

Vector inicial:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 |
|  | |  | |  | |  | |  | |  | |  | |  | |
| 1 | | 3 | | 2 | | 5 | | 6 | | 7 | | -8 | | 9 | |
| x[0] | | x[1] | | x[2] | | x[3] | | x[4] | | x[5] | | x[7] | | x[9] | |
| **X** | | | | | | | | | | | | | | | |

**Código:**

int \*p1, \*p2;  
p1 = x;  
p2 = &X[9];  
\*p2 = (\*p2)\*(-1)-(-1);  
\*p1 = 2\*(\*(p2-1))+\*(p2-2);  
p1++; // p1 = p1 + 1;  
int \*p3 = X + 3;  
\*p3 = \*p3 + 1;  
p2 = p3 + 2;

Vector después de ejecutar el código:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 |
|  | | p1 | |  | | p3 | |  | | p2 | |  | |  | |
| -9 | | 3 | | 2 | | 6 | | 6 | | 7 | | -8 | | -8 | |
| x[0] | | x[1] | | x[2] | | x[3] | | x[4] | | x[5] | | x[7] | | x[9] | |
| **X** | | | | | | | | | | | | | | | |

|  |  |
| --- | --- |
| **Expresión** | **Valor** |
| &p1 | 1000 |
| &p2 | 2000 |
| &p3 | 3000 |
| p1 | 102 |
| p2 | 110 |
| p3 | 106 |
| \*p1 | 3 |
| \*p2 | 7 |
| \*p3 | 6 |
| \*p1 + 1 | 4 |
| \*(p1 + 1) | 2 |
| p3 + 1 | 108 |
| p3 - 2 | 102 |

**Ejercicio 2 - Análisis de código**:

int f(int x, int y, int \*z);  
  
int main() {  
 int a, b, c, d;  
 int \*p1, \*p2 = &a, \*p3 = &d;  
 a = 2;  
 b = -1;  
 p2 = &c;   
 d = f(a + 3, b, &c);  
 p1 = p2;  
 // Parada 1...  
 \*p3 = f(\*p2, -d, p1);  
 // Parada 2...  
 return 0;  
}  
  
int f(int x, int y, int \*z) {  
 \*z = x + y;  
 x++;   
 return \*z + x + y;  
}

**Punto de parada 1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |  |
| ... |  | | | |  |
| 1000 | 2 | | | | a |
| 1004 | -1 | | | | b |
| 1008 | 4 | | | | c, \*p2, \*p1 |
| 1012 | 9 | | | | d, \*p3 |
| .  .  . | .  .  . | | | |  |
|
| 2000 | 1008 | | | | p1 |
| 2004 |
| 2008 | 1008 | | | | p2 |
| 2012 |
| 2016 | 1012 | | | | p3 |
| 2020 |
|  | .  .  . | | | |  |

|  |  |
| --- | --- |
| **Expresión** | **Valor** |
| a | 2 |
| b | -1 |
| c | 4 |
| d | 9 |
| \*p1 | 4 |
| \*p2 | 4 |
| \*p3 | 9 |
| p1 | 1008 |
| p2 | 1008 |
| p3 | 1012 |

**Punto de parada 2:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |  |
| ... |  | | | |  |
| 1000 | 2 | | | | a |
| 1004 | -1 | | | | b |
| 1008 | -5 | | | | c, \*p2, \*p1 |
| 1012 | -9 | | | | d, \*p3 |
| .  .  . | .  .  . | | | |  |
|
| 2000 | 1008 | | | | p1 |
| 2004 |
| 2008 | 1008 | | | | p2 |
| 2012 |
| 2016 | 1012 | | | | p3 |
| 2020 |
|  | .  .  . | | | |  |

|  |  |
| --- | --- |
| **Expresión** | **Valor** |
| a | 2 |
| b | -1 |
| c | -5 |
| d | -9 |
| \*p1 | -5 |
| \*p2 | -5 |
| \*p3 | -9 |
| p1 | 1008 |
| p2 | 1008 |
| p3 | 1012 |

**Ejercicio 3 - Análisis de código:**

#include <stdio.h>  
  
// Codigo tomado de: https://en.wikibooks.org/wiki/C\_Programming/stdio.h/getchar  
  
 int main(void)  
 {  
 char str[1000];  
 int ch, n = 0;  
 while ((ch = getchar()) != EOF && n < 1000) {  
 str[n] = ch;  
 ++n;  
 }   
 for (int i = 0; i < n; ++i)  
 putchar(str[i]);  
  
 putchar('\n');  
 return 0;  
 }

1. **¿Que hace el programa anterior?**

El programa lee caracteres y sólo para cuando haya leído 1000 caracteres o cuando se digite el valor del End-Of-File que en caso de Linux es Ctrl+D o Ctrl+C. Luego imprime todos los caracteres escritos.

1. **Describa las funciones getchar y putchar**

* **getchar():** Es una función en el lenguaje de programación C que lee un solo carácter de la secuencia de entrada stdin estándar, independientemente de lo que sea, y lo devuelve al programa. Se especifica en ANSI-C y es la función de entrada más básica en C. Se incluye en el archivo de encabezado stdio.h.
* **putchar():** La función putchar retorna el carácter escrito. Si ocurre un error de escritura, el indicador de error para el stream es activado y putchar retorna EOF.

1. **¿Cuales son las condiciones necesarias para que el primer ciclo deje de ejecutarse?**

El primer ciclo deja de ejecutarse cuando haya leído 1000 caracteres o cuando se digite el valor del End-Of-File que en caso de Linux es Ctrl+D o Ctrl+C.