

Universidad de Guanajuato División de Ingenierías
Campus Irapuato Salamanca (DICIS)

Algoritmos y estructura de datos
Carlos Hugo García Capulín

Tarea No. 7
Reporte Memoria dinámica

Jair Chávez Islas
14/Octubre/2021

Problema

Tenemos el último programa hecho en clase, el cual, usando memoria dinámica, pide el número de valores de un vector, para luego introducir valor por valor y al final, imprime esos valores y además saca el promedio de los valores del vector.

¿Cómo podríamos hacer para que nos pueda imprimir el mismo vector, pero al inverso?

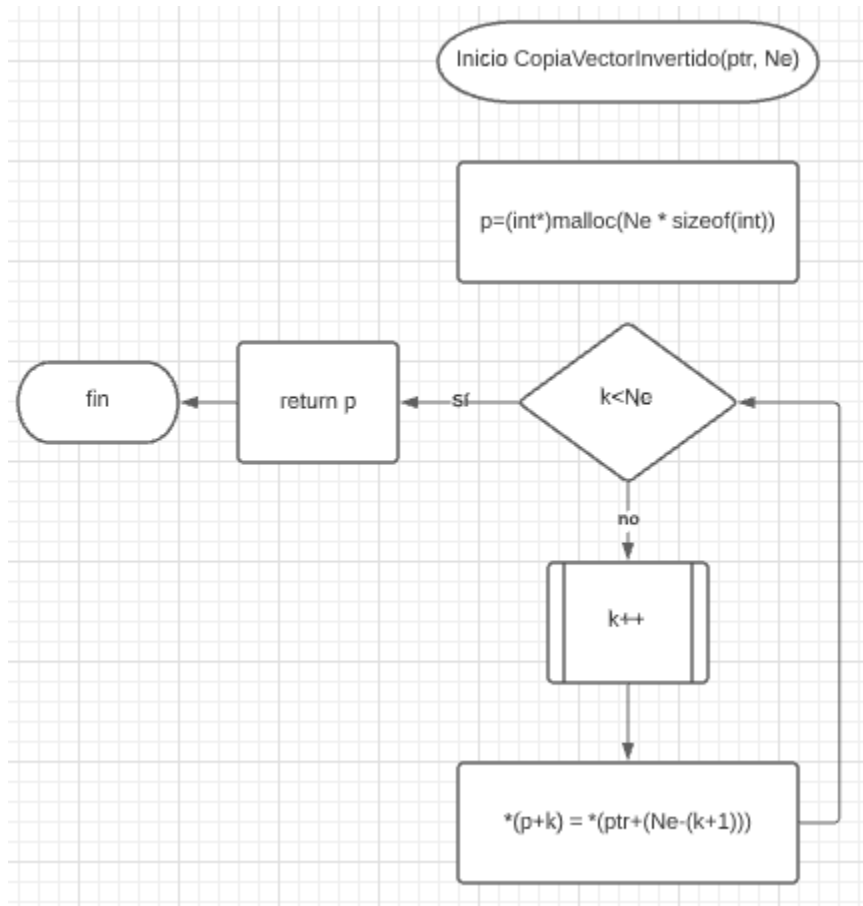
Por ejemplo, si ponemos un vector de 9 valores:

3, 7, 5, 4, 8, 9, 6, 1, 8.

El vector al revés quedaría así: 8, 1, 6, 9, 8, 4, 5, 7, 3.

Solución implementada

Diagrama de la función agregada



Código comentado de la función agregada

```
int* CopiaVectorInvertido(int* ptr, unsigned int Ne)
{
    //declaramos al variable *P
    int *p;

    p=(int*)malloc(Ne * sizeof(int));

    //Se inciaa el ciclo for que nos permite girar los valores del vector
    for(unsigned int k=0;k<Ne;k++)
    {
        *(p+k) = *(ptr+(Ne-(k+1)));
    }

    //Se retorna la variable p
    return p;
}
```

Código completo comentado

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  //Aqui van los prototipos de las funciones
5  int* CapturaVector(unsigned int Ne);
6  void ImprimeVector(int* ptr, unsigned int Ne);
7  float PromedioVector(int* ptr, unsigned int Ne);
8  int* CopiaVectorInvertido(int* ptr, unsigned int Ne);
9
10 //Se inicia la funcion principal
11 int main()
12 {
13     //Se declaran las variables locales de la función principal
14     unsigned int N;
15     int *ptr;
16     float prom;
17     int *vi;
18
19     //Se le pide la cantidad de numeros que tendra el vector
20     printf("Escriba la cantidad de numeros enteros a capturar: ");
21     //Se guarda en la variable N
22     scanf("%u",&N);
23     //Se utilizan las funciones, para la captura, impresion
24     //promedio y la funcion d einvertir el vector
25     ptr=CapturaVector(N);
26     ImprimeVector(ptr,N);
27     prom=PromedioVector(ptr,N);
28     //Con la variable vi, llamamos a la funcion de la copia del vector invertido
29     vi = CopiaVectorInvertido(ptr,N);
30     //Luego la imprimimos con la misma funcion que imprimimos el vector hace un momento
31     ImprimeVector(vi,N);
32
33     //Se libera la memoria
34     free(ptr);
35     free(vi);
36     printf("\n El promedio del vector es: %f",prom);
37
38     //Un salto de linea
39     printf("\n");
40     //Al terminar, si no ha habido errores, retorna un 0
41     return 0;
42 }
43
44 //Se inicia la funcion agregada
45 int* CopiaVectorInvertido(int* ptr, unsigned int Ne)
46 {
47     //declaramos al variable *P
48     int *p;
49
50

```

```

51     p=(int*)malloc(Ne * sizeof(int));
52
53     //Se inciaa el ciclo for que nos permite girar los valores del vector
54     for(unsigned int k=0;k<Ne;k++)
55     {
56         *(p+k) = *(ptr+(Ne-(k+1)));
57     }
58
59     //Se retorna la variable p
60     return p;
61 }
62
63 //Funcion que con el que captura el vector
64 int* CapturaVector(unsigned int Ne)
65 {
66     //Se declaran las variables locales de esta funcion
67     int* puntero;
68     unsigned int k;
69
70     puntero = (int *)malloc(Ne*sizeof(int));
71
72     //Validar si la memoria fue reservada
73     if(puntero==NULL)
74     {
75         printf("Error al reservar la memoria para almacenar los datos");
76         exit(0);
77     }
78
79     //En este ciclo for se repasan el numero de numeros que tendra el vector
80     for(k=0;k<Ne;k++)
81     {
82         //Se piden los numeros del vector al usuario y los guarda en puntero
83         printf("Entero %i: ",k+1);
84         scanf("%i",&puntero[k]);
85     }
86     //se retorna la variable puntero
87     return puntero;
88 }
89
90 //Esta es la funcion para imprimir los vectores

```

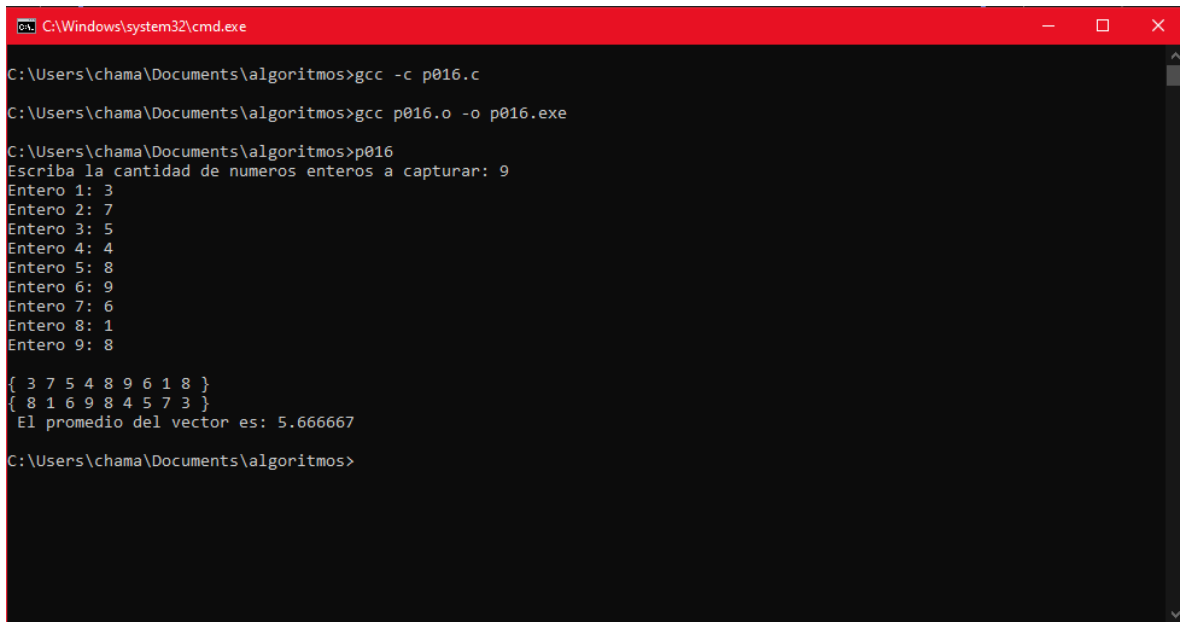
```

90 //Esta es la funcion para imprimir los vectores
91 void ImprimeVector(int* ptr, unsigned int Ne)
92 ▼ {
93     //declaracion de la variable local
94     unsigned int k;
95
96
97     printf("\n{ ");
98     //hace un repaso sobre los numeros del vector apra luego imprimirlos
99     for(k=0;k<Ne;k++)
100         printf("%i ",*(ptr+k));
101     printf("}");
102 }
103
104 //Esta es la funcion para sacar el promedio del vector
105 float PromedioVector(int* ptr, unsigned int Ne)
106 ▼ {
107     //variables locales de la funcion
108     unsigned int k;
109     float result;
110     int sum=0;
111
112     //Se hace un recuento de los numeros del vector
113 ▼ for(k=0; k<Ne; k++)
114     //los suma
115     sum +=*(ptr+k);
116     //guarda en la variable result la division entre sum y Ne
117     result=(float)sum/Ne;
118     //Da el promedio y retorna la variable result
119     return result;
120 }

```

Pruebas y resultados

Evidencia de la función



```
C:\Windows\system32\cmd.exe
C:\Users\chama\Documents\algoritmos>gcc -c p016.c
C:\Users\chama\Documents\algoritmos>gcc p016.o -o p016.exe
C:\Users\chama\Documents\algoritmos>p016
Escriba la cantidad de numeros enteros a capturar: 9
Entero 1: 3
Entero 2: 7
Entero 3: 5
Entero 4: 4
Entero 5: 8
Entero 6: 9
Entero 7: 6
Entero 8: 1
Entero 9: 8

{ 3 7 5 4 8 9 6 1 8 }
{ 8 1 6 9 8 4 5 7 3 }
El promedio del vector es: 5.666667
C:\Users\chama\Documents\algoritmos>
```

Como habíamos previsto, utilizando los mismos valores que en la introducción, pudimos confirmar que los valores al revés son los mismos

Este programa me ayudó mucho a comprender de una mejor manera como se utiliza la memoria dinámica, es un tema en el que me he encontrado algo perdido, pero gracias a el programa pude ejercitar esa parte un poco más.