

4-ALOCAÇÃO DE ARRAYS MULTIDIMENSIONAIS

```
windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/powershell

PS C:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UEOBY\REP_UEOBY\PROG_C\516(alocação_dinamica_memoria)\4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS> cd "c:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UEOBY\REP_UEOBY\PROG_C\516(alocação_dinamica_memoria)\4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS"
PS C:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UEOBY\REP_UEOBY\PROG_C\516(alocação_dinamica_memoria)\4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS> cmd /c .\70.exe

*****
Quantidade de elementos para o vetor: 3
Conteúdo p[0] = 3
Conteúdo p[1] = 6
Conteúdo p[2] = 9
*****

PS C:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UEOBY\REP_UEOBY\PROG_C\516(alocação_dinamica_memoria)\4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS> []
```

```
PROG_C > 516(alocação_dinamica_memoria) > 4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS > C: Tbz > main
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5
6 int main(){//inicio_main
7     printf("\n*****\n");
8
9     int qtd = 3, *p;
10
11     printf("Quantidade de elementos para o vetor: %d\n",qtd);
12
13
14     //alocação de espaço do tipo int
15     p = (int*)malloc(qtd * sizeof(int));
16
17     p[0] = 3;
18     p[1] = 6;
19     p[2] = 9;
20
21     for (int i = 0; i < qtd; i++){
22         printf("Conteúdo p[%d] = %d\n", i, p[i]);
23     }
24
25 }
```

- Quando fazemos esse acesso, já estamos acessando um vetor (ARRAY) UNI-DIMENSIONAL.
- O problema é quando queremos fazer o acesso via multidimensional.

```
516(alocação_dinamica_memoria)\4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS> cmd /c .\70.exe

*****
Conteúdo p[0][0] = 0
Conteúdo p[0][1] = 1
Conteúdo p[0][2] = 2
Conteúdo p[1][0] = 3
Conteúdo p[1][1] = 4
Conteúdo p[1][2] = 5
Conteúdo p[2][0] = 6
Conteúdo p[2][1] = 7
Conteúdo p[2][2] = 8
*****

PS C:\Users\Gabri\Documents\TEORIA_INDIVIDUAL\UEOBY\REP_UEOBY\PROG_C\516(alocação_dinamica_memoria)\4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS> []
```

```
PROG_C > 516(alocação_dinamica_memoria) > 4-ALOCACAO_ARRAYS_MULTIDIMENSIONAIS > C: Tbz > main
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 //
6 * Matriz (3 x 3) 2D-DIMENSIONAL
7 * [00][01][02]
8 * [10][11][12]
9 * [20][21][22]
10 * Alocado para 1 inteiro
11 * int -> 4 bytes -> 1 x 4 = 4 bytes
12 *      2 x 4 = 8 bytes
13 *      3 x 4 = 12 bytes (linhas * colunas * quantidade_de_bytes)
14 *
15
16
17
18 int main(){//inicio_main
19     printf("\n*****\n");
20
21     int *p, linhas = 3, colunas = 3;
22
23     //alocação de espaço do tipo int
24     p = (int*)malloc(linhas * colunas * sizeof(int)); // p = (int*)malloc(colunas * linhas
25
26     for(int i = 0; i < linhas; i++){
27         for (int j = 0; j < colunas; j++){
28             p[i * colunas + j] = 3 * i + j;
29         }
30     }
31
32     for(int i = 0; i < linhas; i++){
33         for (int j = 0; j < colunas; j++){
34             printf("Conteúdo p[%d][%d] = %d\n", i, j, p[i * colunas + j]);
35         }
36     }
37
38     printf("\n*****\n");
39 }
```