



VECTORES

Variável

Um elemento, de um determinado
tipo.

Vector / Array

Conjunto de elementos, de um determinado tipo.

declaração

```
tipo    nome_variavel [n° de elementos]
```

```
float   vencimentos[12];
```

```
int     numeros[100];
```

declaração

```
tipo    nome_variavel [n° de elementos]
```

```
float   vencimentos[12];
```

```
int     numeros[100];
```

inicialização

```
tipo    var[n] = { valor1 , valor2, ... , valorn }
```

```
int     fib[5] = { 1, 2, 3, 5, 8 };
```

declaração

```
tipo nome_variavel [n° de elementos]
```

```
float vencimentos[12];
```

```
int numeros[100];
```

inicialização

```
tipo var[n] = { valor1 , valor2, ... , valorn }
```

```
int fib[5] = { 1, 2, 3, 5, 8 };
```

acesso

```
char vogal[5] = {'h', 'e', 'l', 'l', 'o'};
```

```
vogal[4] = 'X';
```

'h'	'e'	'l'	'l'	'X'
vogal[0]	vogal[1]	vogal[2]	vogal[3]	vogal[4]

Passagem para funções

```
void inicializa(int valores[20]) {  
    ...  
}
```

```
void inicializa(int valores[], int n) {  
    ...  
}
```

Passagem para funções

```
#include "stdio.h"

void inicializa(int valores[], int n);

int main(void) {
    int valores[20], i;
    inicializa(valores, 20);
    for (i = 0; i < 20; i++)
        printf("valores[%d] = %d\n", i, valores[i]);

    return 0;
}

void inicializa(int valores[], int n){
    int i;
    for (i = 0; i < n; i++)
        valores[i] = i;
}
```

Passagem para funções

```
#include "stdio.h"

void inicializa(int valores[], int n);

int main(void) {
    int valores[20], i;
    inicializa(valores, 20);
    for (i = 0; i < 20; i++)
        printf("valores[%d] = %d\n", i, valores[i]);

    return 0;
}

void inicializa(int valores[], int n){
    int i;
    for (i = 0; i < n; i++)
        valores[i] = i;
}
```

```
valores[0] = 0
valores[1] = 1
valores[2] = 2
valores[3] = 3
valores[4] = 4
valores[5] = 5
valores[6] = 6
valores[7] = 7
valores[8] = 8
valores[9] = 9
valores[10] = 10
valores[11] = 11
valores[12] = 12
valores[13] = 13
valores[14] = 14
valores[15] = 15
valores[16] = 16
valores[17] = 17
valores[18] = 18
valores[19] = 19
```

Escreva um programa que guarde os salários pagos a um indivíduo durante M meses.

Em seguida deverá mostrar os valores mensais e o total de todos os meses.

Crie um vector para guardar os salários. Crie uma função que pede ao utilizador o salário de cada mês e depois o guarda no vetor. Crie 3 funções para: mostrar todos os salários, calcular a soma e a média dos salarios. Não use variáveis globais.

Vectores multi-dimensionais

Não existe limite para o número de dimensões que um vector pode ter.

Um vector de duas dimensões não é uma matriz, mas sim um **vector de vectores**.

```
tipo    vector [dim1] [dim2] [...] [dimN]
```

Vectores multi-dimensionais

Não existe limite para o número de dimensões que um vector pode ter;

Um vector de duas dimensões não é uma matriz, mas sim um **vector de vectores**.

```
tipo    vector [dim1] [dim2] [...] [dimN]
```

```
int galo[3][3] = {{0 , 0, 0} , {0, 0, 0} , {0, 0, 0}};
```

0	0	0
0	0	0
0	0	0

Vectores multi-dimensionais

Não existe limite para o número de dimensões que um vector pode ter;

Um vector de duas dimensões não é uma matriz, mas sim um **vector de vectores**.

```
tipo    vector [dim1] [dim2] [...] [dimN]
```

```
int galo[3][3] = {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}};  
galo[0][0] = 1;
```

0	0	0
0	0	0
0	0	0

1	0	0
0	0	0
0	0	0

Vectores multi-dimensionais

Não existe limite para o número de dimensões que um vector pode ter;

Um vector de duas dimensões não é uma matriz, mas sim um **vector de vectores**.

```
tipo    vector [dim1] [dim2] [...] [dimN]
```

```
int galo[3][3] = {{0 , 0, 0} , {0, 0, 0} , {0, 0, 0}};
```

```
galo[0][0] = 1;
```

```
galo[2][1] = 1;
```

0	0	0
0	0	0
0	0	0

1	0	0
0	0	0
0	0	0

1	0	0
0	0	0
0	1	0

Vectores multi-dimensionais

```
int galo[3][3] = {{0 , 0, 0} , {0, 0, 0} , {0, 0, 0}};
```

```
galo          // vector de inteiros com duas dimensões 3x3
```

```
galo[i]       // vector de 3 inteiros
```

```
galo[i][j]    // inteiro na linha i, coluna j do vector
```

Vetores multi-dimensionais

As últimas n-1 dimensões têm de ser especificadas.

```
4  int recebe_vector(int vec[][], int linhas, int cols){
5      return 0;
6  }
7
8  int main()
9  {
10     char vogal2[][] = {{'h', 'e', 'l', 'l'}, {'h', 'e', 'l', 'l'}};
11     return 0;
12 }
```

Vectores multi-dimensionais

As últimas n-1 dimensões têm de ser especificadas.

```
4 int recebe_vector(int vec[][] int linhas, int cols){
5     return 0;
6 }
7
8 int main()
9 {
10     char vogal2[][] = {'h', 'e', 'l', 'l'}, {'h', 'e', 'l', 'l'};
11     return 0;
12 }
```

```
gcc -Wall -g -c main.c -o obj/Debug/main.o
```

```
main.c:4:23: error: array type has incomplete element type
   int recebe_vector(int vec[][], int linhas, int cols){
                        ^
```

```
main.c: In function 'main':
```

```
main.c:10:11: error: array type has incomplete element type
   char vogal2[][] = {'h', 'e', 'l', 'l'}, {'h', 'e', 'l', 'l'};
                ^
```

Vectores multi-dimensionais

As últimas n-1 dimensões têm de ser especificadas.

```
4 int recebe_vector(int vec[][5], int linhas, int cols){
5     return 0;
6 }
7
8 int main()
9 {
10     char vogal2[][5] = {'h', 'e', 'l', 'l'}, {'h', 'e', 'l', 'l'};
11     return 0;
12 }
```

sem erros!

```
gcc -Wall -g -c main.c -o obj/Debug/main.o
main.c: In function 'main':
main.c:10:11: warning: variable 'vogal2' set but not used
[-Wunused-but-set-variable]
    char vogal2[][5] = {'h', 'e', 'l', 'l'}, {'h', 'e', 'l', 'l'};
                   ^
g++ -o bin/Debug/jogo_galo obj/Debug/main.o
Output file is bin/Debug/jogo_galo with size 9,74 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 1 warning(s) (0 minute(s), 0 second(s))
```

Problema

Escreva um programa que guarde os salários pagos a N pessoas durante M meses. Os valores de N e M devem estar especificados numa constante. Use um vector bidimensional.

Em seguida deverá mostrar o total e a média dos salários pagos a cada indivíduo numa tabela.

Crie uma função que pede os salários de um indivíduo. Esta função é chamada pela função que pede todos os salários.

Problema

Exemplo de uma execução do programa.

```
Salarios para pessoa com indice 0
```

```
salario do mes 0: 152.23
```

```
salario do mes 1: 160.50
```

```
-----
```

```
Salarios para pessoa com indice 1
```

```
salario do mes 0: 236.5
```

```
salario do mes 1: 365.5
```

```
-----
```

```
Salarios para pessoa com indice 2
```

```
salario do mes 0: 856.5
```

```
salario do mes 1: 898.35
```

```
-----
```

```
peessoa      total      media
```

```
0    312    156
```

```
1    602    301
```

```
2   1754    877
```

Valores introduzidos pelo utilizador.

Desafio

Jogo do Galo

- vetor bidimensional criado na main guarda o estado do tabuleiro do jogo;
- uma função é responsável pela interação com os jogadores (indica de quem é a vez e pede coordenada para jogar);
- uma função limpa o tabuleiro;
- uma função verifica condições de vitória (pode chamar outras funções), nas linhas, nas colunas, nas diagonais;

Crie outras funções que sejam úteis.

Não usar variáveis globais.