

## Solução Lab 1 – Funções e variáveis indexadas

Ricardo e Myriam

### Exercício 1

```
#include <stdio.h>
#include <stdlib.h>

float media(int n, float nota[]);

int main(int argc, char *argv[])
{
    int i, n;
    float m, nota[20];

    printf("Entre com o numero de notas (max. 20):");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        printf("Entre a nota[%d]:", i+1);
        scanf("%f", &nota[i]);
    }
    m=media(n, nota);
    printf("A media das notas e: %.2f\n", m);

    system("PAUSE");
    return 0;
}
```

```
float media(int n, float nota[])
{
    int i;
    float sum=0;

    for(i=0; i<n; i++)
        sum += nota[i];
    return(sum/n);
}
```

## Solução Lab 1 – Funções e variáveis indexadas

Ricardo e Myriam

### Exercício 2

```
#include <stdio.h>
#include <stdlib.h>

void ordena(int n, float nota[]);

int main(int argc, char *argv[])
{
    int i, n;
    float m, nota[20];

    printf("Entre com o numero de notas (max. 20):");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        printf("Entre a nota[%d]:", i+1);
        scanf("%f", &nota[i]);
    }
    puts("\nNotas nao ordenadas");
    for(i=0; i<n; i++) printf("Nota[%d]: %.2f\n", i+1, nota[i]);

    ordena(n, nota);

    puts("\nNotas ordenadas");
```

```
for(i=0; i<n; i++) printf("Nota[%d]: %.2f\n", i+1, nota[i]);

system("PAUSE");
return 0;
}

void ordena(int n, float nota[])
{
    int i, j;
    float aux;

    for(j=0; j<n-1; j++){
        for(i=0; i<n-1; i++){
            if (nota[i] > nota[i+1]){
                aux=nota[i];
                nota[i]= nota[i+1];
                nota[i+1]=aux;
            }
        }
    }
}
```

## Solução Lab 1 – Funções e variáveis indexadas

Ricardo e Myriam

### Exercício 3

```
#include <stdio.h>
#include <stdlib.h>

void transposta(int n, float matriz[][10], float matriz_t[][10]);

int main(int argc, char *argv[])
{
    int i, j, n;
    float m1[10][10], m1T[10][10];

    printf("Dimensao da matriz quadrada (max. 20):");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            printf("Entre o elemento[%d][%d]:", i+1, j+1);
            scanf("%f", &m1[i][j]);
        }
    }
    puts("\nMatriz original");
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            printf("%.2ft", m1[i][j]);
```

```
        }
        printf("\n");
    }
    transposta(n, m1, m1T);
    puts("\nMatriz transposta");
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            printf("%.2ft", m1T[i][j]);
        }
        printf("\n");
    }
    system("PAUSE");
    return 0;
}

void transposta(int n, float matriz[][10], float matriz_t[][10])
{
    int i, j;
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            matriz_t[i][j]=matriz[j][i];
        }
    }
}
```