

Atividades de Laboratório

– Alocação dinâmica de memória –

1. Crie uma função para preencher os elementos de um vetor v de tamanho n com um certo valor x . Para acessar os elementos do vetor, não utilize índices, use apenas aritmética de ponteiros.

```
void preenche (int *v, int n, int x){  
    //implementar  
}
```

2. Escreva um programa em linguagem C que solicita ao usuário a quantidade de alunos de uma turma e aloca um vetor para armazenar a nota semestral de cada um dos alunos (tipo *float*). Depois de ler as notas e armazená-las no vetor, apresente a média aritmética das notas da turma. O vetor deve ser alocado dinamicamente na memória.
3. Melhore a implementação da questão acima adicionando:
 - (a) uma função para alocar dinamicamente o vetor;
 - (b) uma função que recebe como parâmetro o vetor e devolve a média dos seus elementos.
4. Escreva um programa que guarda num vetor $v1$ (criado dinamicamente) N números inteiros, sendo N previamente especificado pelo utilizador. Em seguida, o programa deve criar um vetor $v2$ com o mesmo tamanho, preenchê-lo com o quadrado dos valores guardados em $v1$, mostrar ambos os vetores na tela (ver exemplo) e liberar a memória alocada.

```
Número de elementos: 4  
Informe o 1o elemento: 1  
Informe o 2o elemento: 2  
Informe o 3o elemento: 3  
Informe o 4o elemento: 10  
Valores inseridos: [ 1 2 3 10 ]  
Valores quadrados: [ 1 4 9 100 ]
```

5. Escreva um programa que leia dois vetores de inteiros com tamanhos definidos pelo utilizador. Escreva uma função que receba como parâmetros os vetores lidos e os respectivos tamanhos e que gere um novo vetor que corresponde à junção dos vetores passados como parâmetro (ver exemplo).

```
Número de elementos do vetor A: 3  
Informe o 1o elemento: 1  
Informe o 2o elemento: 3  
Informe o 3o elemento: 5  
Número de elementos do vetor B: 2  
Informe o 1o elemento: 2  
Informe o 2o elemento: 6  
Resultado: [13526]
```

6. Escreva um programa que leia, para um vetor V , um número N (escolhido pelo usuário) de elementos inteiros. Utilizando funções, construa e exiba dois novos vetores baseados no vetor V : um com os seus números pares e outro com os seus números ímpares (ver exemplo). Todos os vetores deverão ser criados dinamicamente.

```
Número de elementos: 5  
Informe o 1o elemento: 1  
Informe o 2o elemento: 0  
Informe o 3o elemento: 3  
Informe o 4o elemento: 8  
Informe o 5o elemento: 10  
Vetor original: [ 1 0 3 8 10 ]  
Números pares: [ 0 8 10 ]  
Números ímpares: [ 1 3 ]
```

7. Construa uma função que receba, como argumentos duas matrizes e respectivas dimensões (linhas e colunas), determine a sua soma e devolva a matriz resultante da operação, caso seja possível. Se as dimensões das matrizes não permitirem a realização da operação, a função deverá devolver NULL. Construa uma função idêntica que efetue o produto das matrizes (que só é possível quando o número de colunas de uma matriz A for igual ao número de linhas da outra matriz). O programa deverá mostrar o resultado de cada operação ou indicar que não é possível (ver exemplo). Toda a memória alocada deverá ser liberada no final.

```
Número de linhas da matriz A: 2
Número de colunas da matriz A: 3
Informe os elementos da primeira linha: 1 1 1
Informe os elementos da primeira linha: 2 2 2
Número de linhas da matriz B: 2
Número de colunas da matriz B: 3
Informe os elementos da primeira linha: 1 2 3
Informe os elementos da primeira linha: 2 3 4
Resultado da soma:
| 2 3 4 |
| 4 5 6 |
Resultado do produto: não definido.
```

8. Usando um vetor alocado dinamicamente, faça um programa para ler vários números informados pelo usuário e apresentá-los na ordem inversa de sua digitação. O número de valores é desconhecido e a entrada do valor 0 (zero) pelo usuário será o critério de parada. Dica: use um vetor dinâmico e ajuste o tamanho deste vetor para armazenar cada novo valor. O ajuste de tamanho pode ser feito através da função *realloc*.
9. Considerando que na linguagem C uma String é um vetor de caracteres, faça um programa em C para ler uma sequência de caracteres sem espaços, substitua as vogais da sequência pela letra X e apresente a String resultante. Use alocação dinâmica de memória e aritmética de ponteiros para manipular o texto.
10. Implemente a função **f**, chamada na linha 5 do programa abaixo, para que ele imprima os cinco primeiros múltiplos de 3 ao executar as linhas 8 a 10

```
1  int main(){
2  int *v;
3
4  for (int i=0;i<5;i++){
5      v = f(v,i,i*3);
6  }
7
8  for (int i=0;i<5;i++){
9      printf("____");
10     printf("%d ",*(v+i));
11 }
12 }
13
```