



Lista de Exercícios 2 – Vetores Dinâmicos

1. Defina uma situação em que é necessário utilizar vetores dinâmicos.
2. Para que serve a função malloc e a função free?
3. O código abaixo tinha como objetivo guardar 12 inteiros em uma lista que precisa ter tamanho dinâmico. Por que o vetor na verdade só pode armazenar 4 inteiros?

```
int main() {  
    int *vec = malloc(12);  
    ...  
    free(vec);  
    return 0;  
}
```

4. Um erro comum em laços é errar a condição de parada. O trecho abaixo contém um laço for infinito, que armazena na lista o quadrado de sua posição, porém, após algumas dezenas de iterações o programa encerra com um erro. Qual foi o motivo do erro apresentado?

```
int main() {  
    int *vec = malloc(10 * sizeof(int));  
    for(int i = 0; 1; i++) {  
        vec[i] = i * i;  
    }  
    free(vec);  
}
```

Utilize a struct abaixo para os exercícios abaixo.

```
typedef struct TPixel {  
    int red;  
    int green;  
    int blue;  
} TPixel;
```

**Em computação gráfica a tela do computador é representada por uma matriz de pixels, com tamanho determinado pela resolução da tela.**

5. Crie um algoritmo que aloca um vetor da struct TPixel para uma linha que tem tamanho definido pelo usuário.
6. Crie um algoritmo que aloca um vetor de vetores da struct TPixel para uma resolução definida pelo usuário. Ou seja, crie um vetor dinâmico bidimensional para simular uma tela.

Dica: Todos os pixels precisam ser alocados.