

## Alocação Dinâmica de Memória

A alocação dinâmica permite ao programador alocar memória quando o programa está sendo executado.

A “Linguagem C” padrão ANSI define apenas 4 funções para o sistema de alocação dinâmica, disponíveis na biblioteca **stdlib.h**.

## Alocação Dinâmica de Memória

### - malloc

A função **malloc()** serve para alocar memória dinamicamente e tem a seguinte forma:

```
void *malloc (unsigned int num);
```

```
#include <stdio.h>  
#include <stdlib.h>  
main ()  
{  
    int *p, a, i;  
    ...  
    p=(int *)malloc(a*sizeof(int));  
    if (!p)  
    {  
        printf ("** Erro: Memoria Insuficiente **");  
        exit (1);  
    }  
    for (i=0; i<a ; i++)  
        p[i] = i*i;  
    ...  
}
```

## Alocação Dinâmica de Memória - Exercício

Construa um programa que leia da entrada padrão o número de linhas e de colunas de uma matriz de floats, aloque espaço dinamicamente para esta e a inicialize, com valores fornecidos pelo usuário, através da entrada padrão. Ao final o programa deve retornar a matriz na saída padrão com layout apropriado.

## Alocação Dinâmica de Memória

### - **calloc**

A função **calloc()** também serve para alocar memória, mas possui uma forma um pouco diferente:

```
void *calloc (unsigned int num,  
              unsigned int size);
```

A função aloca uma quantidade de memória igual a **num \* size**, isto é, aloca memória suficiente para um vetor de **num** objetos de tamanho **size**.

```
#include <stdio.h>
#include <stdlib.h>
main ()
{
    int *p, a, i;
    ...
    p=(int *)calloc(a,sizeof(int));
    if (!p)
    {
        printf ("** Erro: Memoria Insuficiente **");
        exit (1);
    }
    for (i=0; i<a ; i++)
        p[i] = i*i;
    ...
}
```

## Alocação Dinâmica de Memória - Exercício

Com base no que vimos, construa um programa que aloque dinamicamente memória para um vetor de strings, o número de elementos do vetor e o comprimento máximo das strings pertencentes a este, serão fornecidos pelo usuário, através da entrada padrão. O vetor deve ser inicializado, através da entrada padrão, e posteriormente, impresso na saída padrão.