INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
ENGENHARIA ELETRÔNICA
Programação II

Atividade de Programação – Alocação Dinâmica de Memória

1. Faça um programa que peça ao usuário o tamanho de um vetor de inteiros. Aloque espaço para esse vetor e peça para o usuário entrar com os elementos do vetor. Chame uma função que retorne o somatório dos elementos do vetor, com o seguinte protótipo:
*int sumVector(int *vec, int N);*

2. No mesmo estilo do item anterior, mas agora a função deve retornar um novo vetor com a ordem dos elementos invertida. O vetor também deve ser alocado dinamicamente. Teste de duas formas:
a) *int* revertVector(int *vec, int N);*

b) *void revertVector(int *revec, int *vec, int N);*
Aponte vantagens e possíveis problemas com cada versão.

3. Acesse uma máquina Linux e use Valgrind para avaliar se não há vazamentos de memória (*memory leaks*). De forma resumida, Valgrind funciona da seguinte forma:
valgrind --tool=memcheck --leak-check=yes nome_programa_executavel

Você deve compilar o programa com opção de *debug* (-g), para poder obter informação sobre a linha de código que tem problema.

Teste os seguintes códigos e aponte quais são os problemas.

```
//Código 1
#include <stdlib.h>
int main(void) {
    int *x = malloc(100);
    return 0;
}
```

```
//Código 2
#include <stdlib.h>
int main() {
    int *x = malloc(5);
    x[5] = 3;
    return 0;
}
```

```
//Código 3
#include <stdlib.h>
void f(void) {
  free(x);
}
int main(void) {
    int *x = malloc(100);
    f();
    free(x);
    return 0;
}
```

Teste os códigos que você desenvolveu em 1 e 2.

4. Peça ao usuário o tamanho de duas matrizes e os valores de cada elemento. Armazene os dados de cada matriz em uma estrutura do seguinte tipo:

```
struct _mat {
  int* data;
  int col;
  int row;
}
typedef struct _mat mat;
```

Faça uma função que imprima os elementos da matriz, com o protótipo:
void printMatrix(mat m1);*

Faça uma função para multiplicar as matrizes, observando que isso pode não ser possível, dependendo das dimensões das mesmas (nesse caso, a função deve retornar -1).
Protótipo da função:
int printMatrix(mat m1, mat* m2, mat *mresult);*

Avalie seu código com Valgrind.

Referências
http://www.cprogramming.com/debugging/valgrind.html