

Nome: _____ 2º semestre 2016

Instruções para a prova

- A prova é sem consulta;
- A prova dura 1 hora e 40 minutos;
- Esta folha de enunciados deverá ser entregue ao professor junto com a folha de respostas;
- Onde for adequado, use a função `pow(float x, float y)` para calcular x^y , a função `float sqrt(float x)` para calcular $\sqrt{x}$, a função `float cbrt(float x)` para calcular $\sqrt[3]{x}$, a função `int abs(int x)` para calcular o valor absoluto (módulo) de um número inteiro x , e as funções `float sin(float x)`, `float cos(float x)`, e `float tan(float x)` para calcular respectivamente o seno, cosseno e tangente de um valor de ângulo em radianos.
- Nos exemplos de execução de programas, a saída para a tela emitida pelo programa está em *itálico* e a entrada do usuário está representada em **negrito**.

Questão 1 (50 pontos)

Crie um programa que lê N valores inteiros do teclado e os armazena em um vetor $v1$, e depois lê outros K valores inteiros, que são armazenados em outro vetor $v2$. O programa deve então ler um inteiro indicando uma posição, e inserir o conteúdo de $v2$ em $v1$ a partir dessa posição. O conteúdo de $v1$ não deve ser apagado, mas sim deslocado para a direita a partir da posição definida pelo usuário (veja os exemplos). Os tamanhos dos vetores devem ser fixados via diretivas `#define`. Não esqueça de realizar as validações necessárias. No final, o programa deve imprimir $v1$ com os valores de $v2$ devidamente inseridos, e também o valor do elemento que está na última posição de $v1$.

OBS.: Considere que a função `void imprime_vetor(int vet[], int tamanho)` para mostrar o conteúdo de um vetor JÁ EXISTE. Sua solução deverá apenas CHAMAR esta função onde necessário.

Exemplo de execução (com $N = 7$ e $K = 3$):

```
Digite vetor 1: 0 1 2 3 4 5 6
Digite vetor 2: 50 51 52
Digite posicao: 2
Resultado insercao:
0 1 50 51 52 2 3 4 5 6
Ultimo elemento do vetor: 6
```

Exemplo de execução (com $N = 7$ e $K = 3$):

```
Digite vetor 1: 0 1 2 3 4 5 6
Digite vetor 2: 50 51 52
Digite posicao: 7
Resultado insercao:
0 1 2 3 4 5 6 50 51 52
Ultimo elemento do vetor: 52
```

Exemplo de execução (com $N = 7$ e $K = 3$):

```
Digite vetor 1: 0 1 2 3 4 5 6
```

Digite vetor 2: 50 51 52

Digite posicao: 8

Posicao invalida

Questão 2 (50 pontos)

Escreva um programa em linguagem C++ que deverá ler uma matriz $M \times N$ (M e N estabelecidos via `#define`) de valores inteiros. Em seguida o programa deverá encontrar e imprimir qual é o maior e o menor elemento da matriz. Em seguida o programa deverá aplicar a seguinte regra para cada elemento da matriz: (a) se o elemento for positivo subtraia o maior valor; (b) caso o elemento seja negativo adicione o menor valor; (c) caso seja 0 não faça nada. Depois imprima a matriz resultante.

OBS.: Para a leitura de uma matriz, considere que já existe a função `void ler_matriz(int matriz[][N])`. Para a impressão de uma matriz, considere que já existe a função `void imprime_matriz(int matriz[][N])`. Sua solução deve apenas CHAMAR estas funções onde for apropriado.

Exemplo de execução (com $M=3$, $N=4$):

Digite a matriz inicial:

```
10 1 -5 0
-10 5 -2 0
1 0 -4 5
Menor valor: -10
Maior valor: 10
Nova matriz:
0 -9 -15 0
-20 -5 -12 0
-9 0 -14 -5
```

Nome: _____ 2º semestre 2016

Instruções para a prova

- A prova é sem consulta;
- A prova dura 1 hora e 40 minutos;
- Esta folha de enunciados deverá ser entregue ao professor junto com a folha de respostas;
- Onde for adequado, use a função `float pow(float x, float y)` para calcular x^y , a função `float sqrt(float x)` para calcular $\sqrt{x}$, a função `float cbrt(float x)` para calcular $\sqrt[3]{x}$, a função `int abs(int x)` para calcular o valor absoluto (módulo) de um número inteiro x , e as funções `float sin(float x)`, `float cos(float x)`, e `float tan(float x)` para calcular respectivamente o seno, cosseno e tangente de um valor de ângulo em radianos.
- Nos exemplos de execução de programas, a saída para a tela emitida pelo programa está em *itálico* e a entrada do usuário está representada em **negrito**.

Questão 1 (50 pontos)

Faça um programa que leia um vetor V de inteiros com um número fixo N de elementos indicado via `#define`, e rotacione seus elementos para a direita, fazendo com que o valor de V_{i+1} receba o valor de V_i . Cuidado com as extremidades, o primeiro elemento deve receber o último elemento do vetor. Além disso, o programa deve imprimir o somatório de todos os elementos rotacionados.

OBS.: Considere que a função `void imprime_vetor(int vet[])` para mostrar o conteúdo de um vetor JÁ EXISTE. Sua solução deverá apenas CHAMAR esta função onde necessário.

Exemplo de execução (com $N=6$):

```
Entre vetor:
7 2 3 -1 9 0
Vetor rotacionado:
0 7 2 3 -1 9
Soma dos elementos: 20
```

Questão 2 (50 pontos)

Crie uma função chamada `borrarImagem()`, que recebe uma matriz de inteiros de tamanho $M \times N$ representando uma imagem, e armazena uma versão borrada da imagem original em uma segunda matriz, também de $M \times N$ (utilize `#define` para definir M e N). A versão borrada de um elemento a_{ij} da matriz é dado por $(a_{ij-1} + a_{ij} + a_{ij+1})/3$, onde a é um elemento da matriz original, i indica a linha, e j a coluna. Em outras palavras, estamos tirando a média do elemento com seu vizinho a esquerda e seu vizinho a direita. Tome cuidado com as bordas, sendo que na borda esquerda não há vizinho a esquerda e portanto calculamos $(a_{ij} + a_{ij+1})/2$, e na borda direita não há vizinho a direita, e portanto calculamos $(a_{ij-1} + a_{ij})/2$ como a versão borrada

do elemento. A função `borrarImagem()` também deve retornar o valor médio dos elementos da matriz borrada. A função `borrarImagem()` não deve ler dados do teclado nem imprimir informações na tela. Utilize no programa principal a função que você criou, sendo que você deve imprimir a matriz resultante (borrada) na tela. Note que todos os cálculos são feitos com inteiros, portanto as partes fracionárias são descartadas no exemplo.

OBS.: Para a leitura de uma matriz, considere que já existe a função `void lerMatriz (int matriz[][N])`. Para imprimir os valores da matriz na tela, considere que já existe a função `void imprimirMatriz (int matriz[][N])`. Sua solução deve apenas CHAMAR estas funções onde apropriado.

Exemplo de execução (com $M=4$ e $N=5$):

Digite a matriz da imagem:

```
103 198 105 115 81
255 74 236 41 205
186 171 242 251 227
70 124 194 84 248
```

Imagem borrada:

```
150 135 139 100 98
164 188 117 160 123
178 199 221 240 239
97 129 134 175 166
```

Média elementos da matriz: 157

p3_2015_sols

```
// ----- inserir vetor dentro de outro -----
#include<iostream>
#define N 7
#define K 3
using namespace std;
int main(){
    int vet1[N];
    int vet2[K];
    int vet3[N+K];
    int i,j,k,pos;
    cout<<"Informe o vetor 1: "<<endl;
    for (i=0;i<N;i++){
        cin>>vet1[i];
    }
    cout<<"Informe o vetor 2: "<<endl;
    for (i=0;i<K;i++){
        cin>>vet2[i];
    }
    cout<<"Informe a posição: ";
    cin>>pos;
    if (pos>N) {
        cout<<"Posicao invalida";
    }
    return 0;
}

i=0;
j=0;
while (i<pos){
    vet3[i]=vet1[j];
    i++;
    j++;
}
k=0;
while (k<K){
    vet3[i]=vet2[k];
    i++;
    k++;
}
while (i<N+K){
    vet3[i]=vet1[j];
    i++;
    j++;
}
cout<<"Resultado da insercao: "<<endl;
for (i=0;i<N+K;i++){
    cout<<vet3[i]<<" ";
}
cout<<"Ultimo elemento do vetor e: "<<vet3[N+K-1];
}

//-----some e subtraia o maior/menor valor-----
#include<iostream>
#include<iomanip>
#define M 3
#define N 4
using namespace std;
int main(){
    int mat[M][N]={{10,1,-5,0},{-10,5,-2,0},{1,0,-4,5}};
    int maior=-9999999,menor=9999999,i,j,sentinel;
    for (i=0;i<M;i++){
        for (j=0;j<N;j++){
            if (mat[i][j]>maior) {maior=mat[i][j];}
            if (mat[i][j]<menor) {menor=mat[i][j];}
        }
    }
}
```

```

    }
}
cout<<"Menor valor: "<<menor<<endl;
cout<<"Maior valor: "<<maior<<endl;
for (i=0;i<M;i++){
    for (j=0;j<N;j++){
        sentinela=mat[i][j];
        if (sentinela>0) {
            mat[i][j]=mat[i][j]-maior;
        }
        if (sentinela<0) {
            mat[i][j]=mat[i][j]+menor;
        }
    }
}
for (i=0;i<M;i++){
    for (j=0;j<N;j++){
        cout<<setw(4)<<mat[i][j];
    }
    cout<<endl;
}
}
//-----rotacionar o vetor -----
#include<iostream>
#define N 6
using namespace std;
int main(){
    int vet[N]={7,2,3,-1,9,0};
    int i,salva,soma=0;
    salva=vet[N-1];
    for (i=N-1;i>=0;i--){
        vet[i+1]=vet[i];
    }
    vet[0]=salva;
    cout<<"Vetor rotacionado: "<<endl;
    for (i=0;i<N;i++){
        cout<<vet[i]<<" ";
        soma=soma+vet[i];
    }
    cout<<endl<<"Soma dos elementos :"<<soma;
}
//-----borrar uma imagem digital -----
#include<iostream>
#include<iomanip>
#define M 4
#define N 5
using namespace std;
int borrarimagem(int mo[M][N], int mb[M][N]){
    int i,j,soma;
    for (i=0;i<M;i++){
        mb[i][0]=(mo[i][0]+mo[i][1])/2;
        mb[i][N-1]=(mo[i][N-1]+mo[i][N-2])/2;
    }
    for (i=0;i<M;i++){
        for (j=1;j<N-1;j++){
            mb[i][j]=(mo[i][j-1]+mo[i][j]+mo[i][j+1])/3;
        }
    }
    for (i=0;i<M;i++){
        for (j=0;j<N;j++){
            soma=soma+mb[i][j];
        }
    }
}

```

```
        return soma/(M*N);
    }
int main(){
    int mo[M][N]={103,198,105,115,81},{255,74, 236, 41, 205},
    {186, 171, 242, 251, 227},{70, 124, 194, 84, 248}};
    int mb[M][N];
    int med,i,j;
    med=borrarimagem(mo,mb);
    for (i=0;i<M;i++){
        for (j=0;j<N;j++){
            cout<<setw(4)<<mb[i][j];
        }
        cout<<endl;
    }
    cout<<"Media elementos da matriz: "<<med;
}
```