

Nome: _____ 1º semestre 2014

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 a função `float fabsf(float y)` para calcular o valor absoluto (módulo) de um número real y .
- 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)

Escreva um programa em C++ que leia um vetor de números inteiros de tamanho N (N estabelecido via diretiva `#define`). O programa deve calcular quantas vezes ocorrem valores repetidos em posições consecutivas ("vizinhas") no vetor. Além disso, o programa deve atribuir valor 0 nestas posições do vetor. Ao final, o programa deve exibir na tela o conteúdo do vetor.

OBS.: Para a exibição do conteúdo do vetor na tela, considere que já existe a função `void imprime_vetor(int vet[])`. Sua solução deve apenas chamar esta função.

Exemplo de execução (com $N=10$):

```
Informe vetor:
2 3 3 4 5 1 5 5 7 8
Qtde. de repetições: 2
Vetor final:
2 0 0 4 5 1 0 0 7 8
```

Outro exemplo de execução (com $N=7$):

```
Informe vetor:
2 3 3 3 2 1 5
Qtde. de repetições: 2
Vetor final:
2 0 0 0 2 1 5
```

Questão 1 (50 pontos)

A PROG (Pavês, Rocamboles e Outras Guloseimas) está fazendo uma nova promoção e Teobaldo, que é louco por promoções, quer participar juntando cupons da promoção. Para ter direito a 1 cupom, é preciso juntar embalagens de cada um dos M tipos de produto que a PROG fabrica. Por exemplo, se há 6 tipos de produto e se Teobaldo tiver 3 embalagens de cada um dos produtos P_1, P_2, P_3, P_4 e P_5 , e 2 embalagens do produto P_6 , ele pode obter no máximo 2 cupons da promoção, já que falta uma embalagem do produto P_6 para obter o terceiro cupom.

Escreva um programa em C++ que recebe do usuário M números inteiros (M estabelecido via diretiva `#define`), cada um representando a quantidade de embalagens de cada produto que o usuário tem. O primeiro número neste conjunto representa a quantidade de embalagens do produto P_1 que o usuário possui; o segundo número representa a quantidade de embalagens do produto P_2 , e assim por diante até o produto P_M . Ao final, o programa deve imprimir o número de cupons que o usuário vai obter e quantas embalagens de cada produto restarão depois de obtidos os cupons.

OBS.: Para a exibição do conteúdo do vetor na tela, considere que já existe a função `void mostra_embalagens(int emb[])`. Sua solução deve apenas chamar esta função.

Exemplo de execução: (com $M=6$)

```
Qtde. embalagens: 5 3 4 6 2 2
Qtde. de cupons: 2
Embalagens restantes: 3 1 2 4 0 0
```

Outro exemplo de execução: (com $M=6$)

```
Qtde. embalagens: 10 5 21 3 0 11
Qtde. de cupons: 0
Embalagens restantes: 10 5 21 3 0 11
```

Questão 1 (50 pontos)

Escreva um programa em C++ para ler um vetor de reais de tamanho N (N estabelecido via diretiva `#define`) e um valor X . O programa deve exibir na tela *ACHEI* se o valor X existir pelo menos uma vez no vetor e *NÃO ACHEI* caso contrário. O programa deve também alterar o vetor dividindo todos os seus valores por X . Ao final, o programa deve exibir na tela o conteúdo do vetor.

OBS.: Para a exibição do conteúdo do vetor na tela, considere que já existe a função `void mostra_vetor (float vet[])`. Sua solução deve apenas chamar esta função.

Exemplo de execução: (com $N=6$)

```
Informe vetor: 4 3 12 6 9 3
Informe X: 3
ACHEI.
Vetor final: 1.33 1 4 2 3 1
```

Questão 1 (50 pontos)

Escreva um programa em C++ que leia um vetor de inteiros de tamanho N (N estabelecido via diretiva `#define`) e um valor X . O programa deve calcular o valor que deve ser somado a cada elemento do vetor para que este atinja o valor X . Este resultado deve substituir o valor correspondente no vetor. Além disso, deve ser calculada e exibida na tela a média aritmética dos valores finais do vetor. Ao final, o programa deve exibir na tela o conteúdo do vetor já alterado.

OBS.: Para a exibição do conteúdo do vetor na tela, considere que já existe a função `void mostra_vetor (int vet[])`. Sua solução deve apenas chamar esta função.

Exemplo de execução: (com $N=6$)

```
Informe vetor: 4 3 12 6 9 3
Informe X: 3
Média: -3
Vetor final: -1 0 -9 -2 -6 0
```

Outro exemplo de execução: (com $N=6$)

```
Informe vetor: 42 23 21 36 79 53
Informe X: 50
Média: 7.666
Vetor final: 8 27 29 14 -29 -3
```

Questão 1 (50 pontos)

Escrever um programa em C++ que leia do teclado um vetor ordenado decrescentemente (seu programa não precisa verificar isto) de N números reais (N estabelecido via `#define`), calcule e mostre a média aritmética, a mediana e a diferença entre a média e a mediana dos valores dos elementos do vetor. Para o cálculo da mediana, se o número de elementos do vetor for ímpar, a mediana é o valor do elemento central do vetor; caso par, é dada pela média entre os dois valores dos elementos centrais do vetor. Após a obtenção destes valores, elementos fora do intervalo [média, mediana] ou [mediana, média] (dependendo da ordem das medidas centrais obtidas), devem ter seus valores zerados.

OBS.: O vetor alterado deverá ser mostrado na tela utilizando a função com o protótipo `void imprime (float v[])`; , que não precisa ser codificada em sua resposta. Sua solução deve apenas chamar esta função.

Exemplo de execução: (com $N=8$)

```
Vetor:
10.5 9.8 8 9.4 7.3 6 5.2 5
Média aritmética: 7.65
Mediana: 8.35 (= (9.4 + 7.3) / 2)
Média - mediana: -0.70
Vetor alterado:
0 0 8 0 0 0 0 0
```

Outro exemplo de execução: (com $N=9$)

```

Vetor:
9.9 9.8 9.4 8 7.3 6 5.2 5 4.9
Média aritmética: 7.344
Mediana: 7.3
Média - mediana: 0.044
Vetor alterado:
0 0 0 0 7.3 0 0 0 0

```

Questão 2 (50 pontos)

Escreva um programa em C++ que deverá ler uma matriz com M linhas x N colunas (M e N estabelecidos via `#define`) de valores inteiros, onde cada elemento da matriz representa uma cota (em centímetros) do correspondente metro quadrado de um terreno de $M \times N$ metros que precisa ser aplainado. Ao final, o programa deve imprimir, como no exemplo abaixo, o volume de terra (em metros cúbicos) que será necessário adicionar (ou retirar) para fazer a terraplenagem de toda a área do terreno, visando colocar todo o terreno em uma cota uniforme de 128 centímetros.

OBS.: Para a leitura de uma matriz, considere que já existe a função `void ler_matriz (int matriz[][N])`. Sua solução deve apenas chamar esta função.

Exemplo de execução (com $M=3$ e $N=6$):

```

Informe a matriz original:
19 68 58 80 21 177
66 28 30 32 34 156
78 58 60 52 74 172
Qtde. de terra a acrescentar: 10.41 m3

```

Outro exemplo de execução (com $M=3$ e $N=6$):

```

Informe a matriz original:
219 168 158 180 125 217
166 128 130 232 234 136
278 158 160 152 214 152
Qtde. de terra a retirar: 9.03 m3

```

Questão 2 (50 pontos)

Escreva um programa em linguagem C++ que deverá ler uma matriz com M linhas x N colunas (M e N estabelecidos via `#define`) de valores reais **não-nulos**[]. Em seguida, o programa deve **criar uma nova matriz**[alterar a matriz] de forma a dividir os valores dos elementos de cada linha i da matriz **original** [] pelo valor do elemento da coluna i da mesma linha e, no caso de não existir a coluna, a divisão deve ser pelo último valor da mesma linha. Ao final, o programa deve mostrar na tela a matriz resultante e a quantidade de valores na matriz resultante que estão acima de 1 (um).

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

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

```

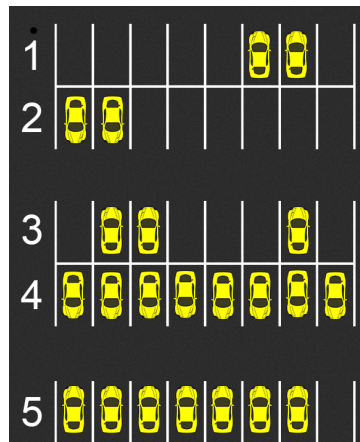
Informe a matriz:
37 28 41
17 41 53
25 93 31
93 19 26
Matriz resultante[alterada]
1.000 0.757 1.108
0.414 1.000 1.293
0.806 3.000 1.000
3.577 0.731 1.000
4 valores acima de 1.

```

Questão 2 (50 pontos)

Um estacionamento quer controlar seu movimento de carros e instalou sensores em cada uma de suas vagas.

Os sensores informam, sempre que requisitado, se existe ou não um veículo estacionado na vaga, resultando em uma matriz com valores que representam vaga livre ou vaga ocupada.



Escreva um programa em linguagem C++ que receba uma matriz representando um estacionamento de M corredores, cada corredor com N vagas (M e N estabelecidos via `#define`), cada elemento da matriz assumindo valor 1 (vaga ocupada) ou 0 (vaga livre). O programa deve informar: (i) se existe ou não algum corredor totalmente ocupado (se existir, informar quais), e (ii) quantas vagas do estacionamento estão ocupadas e quantas estão livres.

OBS.: Para a leitura de uma matriz, considere que já existe a função `void ler_estacionamento (int matriz[][N])`. Sua solução deve apenas chamar esta função.

Exemplo de execução (com $M=5$ e $N=8$):

Informe estacionamento:

```
0 0 0 0 0 1 1 0
1 1 0 0 0 0 0 0
0 1 1 0 0 0 1 0
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0
```

Corredor 4 está lotado.

Existem 22 vagas ocupadas e 18 livres.

Questão 2 (50 pontos)

Teobaldo é proprietário de uma concessionária de carros e precisa da sua ajuda para verificar a produtividade de N funcionários das vendas semanais de um mês. Sua tarefa é escrever um programa em C++ que receba do usuário uma matriz de N linhas x 4 colunas (N estabelecido via `#define`) de valores reais, na qual cada linha representa um vendedor e as colunas são os totais (em reais) de vendas feitas por ele em cada semana do mês (considerando-se meses de 4 semanas). Ao final, seu programa deve mostrar na tela: a) o vendedor mais produtivo, b) o total do mês, c) a média dos vendedores.

OBS.: Para a leitura de uma matriz, considere que já existe a função `void ler_matriz (float matriz[][4])`. Sua solução deve apenas chamar esta função.

Exemplo de execução (com $N=3$):

Informe as vendas:

```
30000 25450 12500 10000
11500 13550 21450 12999
40000 35000 15600 19500
```

Vendedor mais produtivo: 3

Total Mensal: R\$ 247549.00

Média dos Vendedores: R\$ 82516.33

Questão 2 (50 pontos)

O município de Blumenau, em Santa Catarina, para proteger os cidadãos de calamidades devidas a enchentes, monitora o nível (em metros) do rio Itajaí-Açu de tempos em tempos, e criou um sistema de alerta à população.

Todos os locais do município suscetíveis a enchentes tiveram suas cotas (em metros), em relação ao fundo do rio, mapeados e registrados. Foram também identificados e catalogados abrigos para os quais a população deve deslocar-se em caso de enchente.

Para cada local do município sujeito a enchentes foram registrados o código do local, sua cota (em metros) em relação ao fundo do rio, o código do abrigo para onde seus moradores devem deslocar-se em caso de enchente e o número de avisos emitidos para aquele local (inicialmente com valor zero e incrementado de 1 a cada processamento do programa com situação de enchente detectada).

Escrever um programa em C++ que leia do usuário o nível atual do rio e uma matriz de monitoramento de 4 linhas e N colunas (N estabelecido via #define), cada coluna correspondente aos dados dos locais do município sujeitos a enchentes e cada linha correspondente a código do local, cota, código de abrigo e número de avisos, conforme descrito anteriormente. O programa deve mostrar, para os locais com cotas em situação de enchente, os locais, cotas e abrigos e, ainda, incrementar o número de avisos emitidos para aquele local na matriz.

Exemplo de execução (para N = 9):

Nível do rio (m): 10.05

Matriz de monitoramento:

30	31	32	33	34	35	36	37	40
9	9.3	9.5	9.7	9.5	11	12.3	10.9	11
90	90	90	91	91	92	55	55	55
3	2	2	1	2	2	1	0	0

Enchentes:

Locais	Cotas	Abrigos
30	9	90
31	9.3	90
32	9.5	90
33	9.7	91
34	9.5	91

Matriz de monitoramento final:

30	31	32	33	34	35	36	37	40
9	9.3	9.5	9.7	9.5	11	12.3	10.9	11
90	90	90	91	91	92	55	55	55
4	3	3	2	3	2	1	0	0

p3_2014s1_sols

```
//----- 1-----
#include<iostream>
#define N 10
using namespace std;
void imprime_vetor(int v[]){
    int i;
    for (i=0;i<N;i++){
        cout<<v[i]<<' ';
    }
}

int main() {
    int vet[N]={2,3,3,4,5,1,5,5,7,8};
    // int vet[N]={1,3,3,3,3,3,3,3};
    int i,j,k,ctd;
    // for (i=0;i<N;i++){
    //     cout<<"Informe "<<i+1<<" ";
    //     cin>>vet[i];
    // }
    ctd=0;
    i=0;
    while(i<N){
        j=1;
        while ((vet[i]==vet[i+j])&&(i+j<N)){
            j=j+1;
        }
        ctd=ctd+j-1;
        if (j!=1){
            k=0;
            while (k<j){
                vet[i]=0;
                k++;
                i++;
                cout<<"valor de i "<<i<<endl;
            }
            i--; // foi incluído
        }
        i++;
        cout<<"valor de i fora " <<i<<endl;

    }
    cout<<"Quantidade de repeticoes " <<ctd<<endl;
    cout<<"Vetor final: " <<endl;
    imprime_vetor(vet);
}

//-----2-----
#include<iostream>
#define M 6
using namespace std;
void imprime(int vx[]){
    int j;
    for (j=0;j<M;j++){
        cout<<vx[j]<<' ';
    }
}

int main() {
    int i,j,men;
    int v[M];
    for (i=0;i<M;i++){
        cout<<"Informe "<<i+1<<' ';
        cin>>v[i];
    }
}
```

```

    men=99999;
    for (i=0;i<M;i++){
        if (v[i]<men){
            men=v[i];
        }
    }
    cout<<"Quantidade de embalagens: ";
    imprime(v);
    cout<<endl;
    cout<<"Quantidade de cupons: ";
    cout<<men<<endl;
    cout<<"Embalagens restantes: ";
    for (i=0;i<M;i++){
        v[i]=v[i]-men;
    }
    imprime(v);
    cout<<endl;
}
//----- 3-----
#include<iostream>
#define M 6
using namespace std;
void imprime(float vx[]){
    int j;
    for (j=0;j<M;j++){
        cout<<vx[j]<<' ';
    }
}
int main() {
    int i,j,achei;
    float X;
    float v[M];
    for (i=0;i<M;i++){
        cout<<"Informe "<<i+1<<' ';
        cin>>v[i];
    }
    cout<<"Informe X ";
    cin>>X;
    achei=0;
    for (i=0;i<M;i++){
        if (v[i]==X){
            achei=1;
        }
    }
    if (achei==1){
        cout<<"ACHEI"<<endl;;
    }
    else {
        cout<<"NAO ACHEI"<<endl;
    }
    for (i=0;i<M;i++){
        v[i]=v[i]/X;
    }
    imprime(v);
}
//----- 4 -----
#include<iostream>
#define M 6
using namespace std;
void imprime(float vx[]){
    int j;
    for (j=0;j<M;j++){
        cout<<vx[j]<<' ';

```

```

    }
}
int main() {
    int i;
    float v[M];
    float som, X;
    for (i=0;i<M;i++){
        cout<<"Informe "<<i+1<<" ";
        cin>>v[i];
    }
    cout<<"Informe X ";
    cin>>X;
    for (i=0;i<M;i++){
        v[i]=X-v[i];
    }
    som=0;
    for (i=0;i<M;i++){
        som=som+v[i];
    }
    som=som/M;
    cout<<"Media: "<<som<<endl;
    cout<<"Vetor final: ";
    imprime(v);
}
// ----- 5-----

//----- 6-----
#include<iostream>
#define n 9
using namespace std;
void impvet(float x[n]){
    int i,j;
    for (i=0;i<n;i++){
        cout<<x[i]<<" ";
    }
}
int main(){
    float v[n];
    int i,j;
    float soma,media,mediana;
    soma=0;
    for (i=0;i<n;i++){
        cout<<"informe "<<i+1<<" ";
        cin>>v[i];
        soma=soma+v[i];
    }
    if (n%2==0){
        i=(n/2)-1;
        j=i+1;}
    else {
        i=j=n/2;
    }
    media=soma/n;
    mediana=(v[i]+v[j])/2;
    if (media>mediana){
        swap(media,mediana);
    } // agora mediana,media esta em ordem...
    for (i=0;i<n;i++){
        if ((v[i]<mediana)|| (v[i]>media)){
            v[i]=0;
        }
    }
}
cout<<"Media aritmetica: "<<media;

```

```

    cout<<"Mediana: "<<mediana;
    cout<<"Media - Mediana: "<<media-mediana;
    cout<<"Vetor alterado: "<<endl;
    impvet(v);
}

//----- 7-----
#include<iostream>
#define m 3
#define n 6
using namespace std;
void impvet(float x[m][n]){
    int i,j;
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            cout<<x[i][j]<<' ';
        }
        cout<<endl;
    }
}

int main(){
//    float v[m][n]={19,68,58,80,21,177},{66,28,30,32,34,156},{78,58,60,52,74,172}};
    float v[m][n]={219,168,158,180,125,217},{166,128,130,232,234,136},
    {278,158,160,152,214,152}};
    float qtd;
    int i,j;
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            if (v[i][j]>128){
                qtd=qtd+((128-v[i][j])/100);
            }
            else{
                qtd=qtd-((v[i][j]-128)/100);
            }
        }
    }
    impvet(v);
    if (qtd>0){
        cout<<"Qtdade de terra a acrescentar: "<<qtd<<endl;}
    else{
        cout<<"Qtdade de terra a retirar      : "<<-qtd<<endl;
    }
}

//----- 8-----
#include<iostream>
#include<iomanip>
#define m 4
#define n 3
using namespace std;
void impvet(float x[m][n]){
    int i,j;
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            cout<<setw(6)<<setprecision(4)<<x[i][j]<<' ';
        }
        cout<<endl;
    }
}

void levet(float x[m][n]){
    int i,j;
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){

```

```

        cout<<"Informe l= "<<i+1<<" c= "<<j+1<<' ';
        cin>>x[i][j];
    }
}

int main(){
    float en[m][n]={37,28,41},{17,41,53},{25,93,31},{93,19,26}};
    float sa[m][n];
    int i,j,js,qtd;
    qtd=0;
//    levet(en);
    impvet(en);
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            if (i>=m-1){
                js=n-1; }
            else{
                js=i;
            }
            sa[i][j]=en[i][j]/en[i][js];
            if (sa[i][j]>1.000){
                qtd++;}
        }
    }
    impvet(sa);
    cout<<"Quantidade "<<qtd<<" valores acima de 1."<<endl;
}

//----- 9-----
#include<iostream>
#include<iomanip>
#define m 5
#define n 8
using namespace std;
void impvet(float x[m][n]){
    int i,j;
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            cout<<setw(6)<<x[i][j]<<' ';
        }
        cout<<endl;
    }
}

void levet(float x[m][n]){
    int i,j;
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            cout<<"Informe l= "<<i+1<<" c= "<<j+1<<' ';
            cin>>x[i][j];
        }
    }
}

int main(){
    int en[m][n]={0,0,0,0,0,1,1,0},{1,1,0,0,0,0,0,0},
                {0,1,1,0,0,0,1,0},{1,1,1,1,1,1,1,1},
                {1,1,1,1,1,1,1,0}};
    int i,j,livre,qtdvo,qtdvl;

    qtdvo=qtdvl=0;
    // corredor livre
    for (i=0;i<m;i++){
        livre=0;
        for (j=0;j<n;j++){

```

```

        if (en[i][j]==1){

            qtdvo++;

        }
        else {
            qtdvl++;

            livre=1;

        }
    }
    if (livre==0){
        cout<<"Corredor "<<i+1<<" lotado"<<endl;
    }

}

cout<<"Existem "<<qtdvo<<" vagas ocupadas e "<<qtdvl<<" vagas livres";
}
//----- 10 -----
#include<iostream>
#include<iomanip>
#define n 3
using namespace std;
void impvet(float x[n][4]){
    int i,j;
    for (i=0;i<n;i++){
        for (j=0;j<4;j++){
            cout<<setw(6)<<setprecision(4)<<x[i][j]<<' ';

        }
        cout<<endl;
    }
}

void levet(float x[n][4]){
    int i,j;
    for (i=0;i<n;i++){
        for (j=0;j<4;j++){
            cout<<"Informe l= "<<i+1<<" c= "<<j+1<<' ';
            cin>>x[i][j];
        }
    }
}

int main(){
    float en[n][4]={{30000,25450,12500,10000},{11500,13550,21450,12999},
    {40000,35000,15600,19500}};
    // levet(en);
    float mx=-9999;
    float total,media,som;
    int i,j,qual;
    total=0;
    media=0;
    for (i=0;i<n;i++){
        som=0;
        for(j=0;j<4;j++){
            som=som+en[i][j];
            total=total+en[i][j];
            media=media+en[i][j];
        }
        if (som>mx){
            mx=som;
            qual=i+1;
        }
    }
    cout<<"Vendedor mais produtivo: "<<qual<<endl;
    cout<<"Total mensal: R$ "<<total<<endl;
}

```

```

        media=media/n;
        cout<<"Media dos vendedores: R$ "<<media<<endl;
    }
//-----
#include<iostream>
#include<iomanip>
#define N 9
using namespace std;
int main() {
    float en[4][9]={30,31,32,33,34,35,37,40},{9,9.3,9.5,9.7,9.5,11,12.3,10.9,11},
                {90,90,90,91,91,92,55,55,55},{3,2,2,1,2,2,1,0,0}};

    float rio;
    int i,j;
    cout<<"Informe o nivel do rio em m ";
    cin>>rio;
    for (i=0;i<N;i++){
        if (en[1][i]<rio){
            cout<<en[0][i]<<"      "<<en[1][i]<<"      "<<en[2][i]<<endl;
            en[3][i]++;
        }
    }
    cout<<"Matriz de monitoramento final"<<endl;
    for (j=0;j<4;j++){
        for (i=0;i<N;i++){
            cout<<setw(5)<<en[j][i]<<' ';
        }
        cout<<endl;
    }
}

```