CI208 - Programação de Computadores — Prova 3

Nome: _____ 1º semestre 2015

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)

Considere um programa que gerencia uma prateleira de locadora de filmes. A prateleira é representada no programa por um vetor, onde cada posição equivale a um filme e à quantidade de exemplares disponíveis para a locação. Por exemplo, o número 3 na posição 7 indica que existem 3 exemplares disponíveis do filme que está na posição 7. O valor 0 indica que não há exemplar disponível para ser alugado onde quer que ele esteja.

Escreva um programa em C++ que leia N quantidades de exemplares (N estabelecido via `#define`), e que depois de efetuar a leitura da situação atual da prateleira, gerencie a locação e devolução dos filmes da seguinte forma: o usuário deverá digitar `f` para finalizar, `a` para indicar que será uma locação, e `d` para indicar uma devolução. Após selecionar `a` ou `d`, o usuário deverá digitar uma sequência de tuplas representando a posição do filme na prateleira e a quantidade de filmes a serem locados ou devolvidos. O valor `-1` encerra esta leitura. A cada etapa de aluguel ou devolução, o programa deverá mostrar a situação da prateleira até que o usuário digite a opção `f` para finalizar a execução. Se não houver quantidade suficiente para a venda, esta não deve ser feita e a prateleira fica sem mudanças.

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=5):

```
Entre situação da prateleira:
2 4 0 1 2
(a)lugar? (d)evolver ou (f)inalizar? a
0 1
Alugado filme 0.
2 1
Filme 2 sem exemplares.
3 1
Alugado filme 3.
-1 -1
Situação da prateleira: 1 4 0 0 2
(a)lugar? (d)evolver ou (f)inalizar? d
2 2
Devolvido filme 2.
3 1
Devolvido filme 3.
-1 -1
Situação da prateleira: 1 4 2 1 2
(a)lugar? (d)evolver ou (f)inalizar? a
4 4
Exemplares do filme 4 insuficientes.
-1 -1
Situação da prateleira: 1 4 2 1 2
(a)lugar? (d)evolver ou (f)inalizar? f
Fim da execução.
```

Questão 2 (50 pontos)

Faça um programa em C++ que leia uma matriz $M \times N$ de números inteiros (M e N estabelecidos via `#define`), calcule e imprima quantos elementos 2-maior existem nessa matriz. Um elemento é 2-maior caso seja maior que os seus vizinhos de linha (da direita e da esquerda). Caso não exista vizinho, considere que este possui o valor 0 (zero). Para resolver o problema proposto pede-se o desenvolvimento e uso de uma função de nome `calcula_2maior()` que recebe como entrada uma matriz, calcula e retorna o número de elementos 2-maior.

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, N=4):

```
Informe a matriz
7   15   4    11
5   3    20    0
1   6    9     8
Qtde. elementos 2-maior: 5
```

CI208 - Programação de Computadores — Prova 3

Nome: _____ 1º semestre 2015

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++ para ler um vetor de N elementos inteiros (N estabelecido via `#define`). Em seguida, pergunte ao usuário qual posição deve ser eliminada. A partir dessa posição o vetor terá seus elementos deslocados uma posição à esquerda (na direção do inicio do vetor). A posição vaga no final do vetor após o deslocamento devem receber valor 0 (zero). Ao final, o programa deve imprimir o valor do elemento excluído e o conteúdo do vetor resultante.

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

Exemplo de execução (com N=10):

```
Informe vetor:
 5 6 8 2 1 3 9 4 7 10
Indique posicao a eliminar: 4
Vetor final: 5 6 8 2 3 9 4 7 10 0
Valor eliminado: 1
```

Questão 2 (50 pontos)

Escrever um programa em C++ que leia uma matriz de número reais de ordem $M \times N$ (M e N estabelecidos via *#define*, sendo $N > 2$), verifique e mostre as linhas nas quais os dois primeiros valores da linha e qualquer dos valores restantes da mesma linha podem formar um triângulo. Indicar os casos de linhas que não permitam a formação de triângulo algum. Para a fomação de um triângulo, a soma de dois lados quaisquer deve ser maior que o terceiro.

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

Exemplo de execução (com M=5, N=4):

```
Indique a matriz:
3 4 5 8
0 0 0 0
1 2 3 4
4 3 2 3
1 1 1 3
Resultado:
Linha 0 colunas 0 e 1 com 2
Linha 1 não permite formar triângulos
Linha 2 não permite formar triângulos
Linha 3 colunas 0 e 1 com 2 e 3
Linha 4 colunas 0 e 1 com 2
```

CI208 - Programação de Computadores — Prova 3

Nome: _____ 1º semestre 2015

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)

Uma companhia de eletricidade implantou uma política que reajusta a conta de energia elétrica daqueles que gastam além de um determinado valor limite. O reajuste varia de acordo com uma das seguintes situações: (i) reajuste de 10% para quem gastou até 40% além do valor limite; (ii) reajuste de 25% para quem gastou de acima de 40% e até 70% do valor limite; (iii) reajuste de 50% para quem gastou acima de 70% do valor limite.

Faça um programa em C++ que obtenha um valor limite para cálculos de reajustes, e um vetor de N valores reais (N estabelecido via `#define`), cada valor representando o valor gasto por cada uma das N residências avaliadas. Atualize o valor das contas de cada residência no vetor caso elas ultrapassem o valor limite. Por fim, mostre os valores reajustados que cada residência deve pagar para saldar sua conta de energia e o total acumulado que será arrecadado a mais pela companhia elétrica.

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

Exemplo de execução (com N=10):

```
Digite valor limite: 100
Digite contas de residencias:
20 40 60 80 100 120 140 160 180 200
Valores antes do reajuste:
20 40 60 80 100 120 140 160 180 200
Valores depois do reajuste:
20 40 60 80 100 132 154 200 270 300
Valor extra arrecadado: 256
```

Exemplo de execução (com N=5):

```
Digite valor limite: 300
Digite contas de residencias:
100 200 300 400 500
Valores antes do reajuste:
100 200 300 400 500
Valores depois do reajuste:
100 200 300 440 625
Valor extra arrecadado: 165
```

Questão 2 (50 pontos)

Faça um programa em C++ que leia uma matriz $M \times N$ de números inteiros (M e N estabelecidos via `#define`) e: imprime **Borda** se a soma dos elementos da borda da matriz é maior que soma do elementos do miolo da matriz; ou imprime **Miolo** caso contrário. A borda da matriz são todos os elementos das primeiras e últimas colunas e linhas; o miolo são todos os outros elementos da matriz. Você deve escrever a função `MioloMoleCascaDura()`, que recebe uma matriz $M \times N$ e retorna 1 (um) se a soma dos elementos da borda da matriz é maior ou igual do que a soma do elementos do miolo e 0 (zero) caso contrário.

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, N=4):

```
Informe matriz:
1  4 2 5
23 5 4 9
9  3 7 5
Borda
```

Outro exemplo de execução (com M=3, N=4):

```
Informe matriz:
1  4  1 5
1 25 14 2
2  3  7 5
Miolo
```

CI208 - Programação de Computadores — Prova 3

Nome: _____ 1º semestre 2015

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)

Faça um programa em C++ que leia um vetor de tamanho N (N estabelecido via `#define`) contendo inteiros $x_1, x_2, \ldots, x_N$ representando o resultado da rodada de um campeonato de futebol com N times. Cada valor x_k pode ser: (a) **negativo** (ou nulo), representando a diferença de gols pela qual o time k perdeu (ou empatou) o jogo dessa rodada; (b) **positivo**, representando o índice (de 1 a N) do time que foi adversário do time k nessa rodada. Por exemplo, para sequência -2 1 4 0, temos que: o time 1 perdeu o jogo por uma diferença de 2 gols, o time 2 jogou contra o time 1, o time 3 jogou contra o time 4 e o time 4 empatou a partida. Seu programa deve alterar o vetor lido de forma que toda posição passe a ter o saldo de gols do respectivo time nessa rodada. Para o exemplo, o 2º elemento deve ser alterado para 2 (pois o time 2 jogou contra o time 1 que tem um saldo de -2 gols) e o 3º elemento deve ser alterado para 0 (pois o time 3 jogou contra o time 4 que tem um saldo de 0 gols). Ao final, seu programa deve imprimir o vetor resultante, a maior diferença de gols que houve nas partidas da rodada e a quantidade de partidas que terminaram empatadas. Considere que todo time perdedor apresenta no vetor o seu saldo de gols e todo time ganhador apresenta o índice do seu adversário, variando de 1 a N. No caso de empate, apenas um deles tem o índice do adversário.

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):

Resultado da rodada:
-2 6 1 5 -1 0
Saldo de gols:
-2 0 2 1 -1 0
Maior diferença de gols: 2
Qtd. de jogos empatados: 1

Outro exemplo de execução (com N=10):

Resultado da rodada:
0 0 -3 9 1 -3 6 2 -1 3
Saldo de gols:
0 0 -3 1 0 -3 3 0 -1 3
Maior diferença de gols: 3
Qtd. de jogos empatados: 2

Questão 2 (50 pontos)

Faça um programa em C++ que leia uma matriz quadrada $M \times M$ de números inteiros (M estabelecido via `#define`), calcule e imprima a soma dos menores elementos de cada linha e de cada coluna da matriz. Crie e utilize no programa a função `smenorColLin()`, que recebe uma matriz $M \times M$ e um valor k e retorna a soma do menor elemento da linha k com o menor elemento da coluna k da matriz.

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

Exemplo de execução (com M=4):

Informe matriz:
1 4 2 5
23 5 4 9
9 3 7 5
6 5 4 2
Soma dos menores: 18

CI208 - Programação de Computadores — Prova 3

Nome: _____ 1º semestre 2015

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)

Faça um programa em C++ que leia (do teclado) um vetor de tamanho N (N estabelecido via `#define`) de números inteiros e altere este vetor, de forma que todos os elementos divisíveis pelo primeiro valor do vetor recebam o novo valor -1. Ao final do processamento, o programa deverá mostrar o novo conteúdo do vetor no monitor de vídeo e informar a soma dos valores que foram substituídos por -1.

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

Exemplo de execução (com N=10):

Informe vetor:
3 4 2 5 6 3 7 9 12 15
Vetor gerado:
-1 4 2 5 -1 -1 7 -1 -1 -1
Soma: 48

Questão 2 (50 pontos)

Fazer programa em C++ que lê uma matriz $M \times N$ (M e N estabelecidos via diretiva *#define*) a partir do teclado, e informa se a matriz lida atende o Modelo de Vandermonde Totalmente, Parcialmente, ou não Atende. A avaliação deve ser feita por uma função chamada `Avalia_Vandermonde` (que deverá ser codificada junto com o resto do código) que recebe a matriz e retorna o valor 1 (indicando que atende Totalmente), o valor 0 (indicando que atende Parcialmente) ou o valor -1 (indicando que não Atende Vandermode).

Uma matriz atende totalmente o Modelo de Vandermonde quando M for igual a N e cada item da matriz está dentro do Modelo abaixo. Uma matriz atende parcialmente o modelo de Vandermonde quando está dentro do modelo abaixo, mas $M \neq N$. E a matriz não atende o modelo de Vandermonde quando não atende o modelo abaixo com quaisquer dos valores da matriz.

No modelo abaixo, $X_0, X_1, \cdots X_{M-1}$ são os valores básicos da matriz e podem possuir valores repetidos (por exemplo, X_0 pode ser igual a X_6).

$$\begin{vmatrix} 1 & {X_0}^1 & {X_0}^2 & {X_0}^3 & {X_0}^4 & \cdots & {X_0}^{N-1} \\ 1 & {X_1}^1 & {X_1}^2 & {X_1}^3 & {X_1}^4 & \cdots & {X_1}^{N-1} \\ 1 & {X_2}^1 & {X_2}^2 & {X_2}^3 & {X_2}^4 & \cdots & {X_2}^{N-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & {X_{M-1}}^1 & {X_{M-1}}^2 & {X_{M-1}}^3 & {X_{M-1}}^4 & \cdots & {X_{M-1}}^{N-1} \end{vmatrix}$$

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, N=3):

Digite matriz 3x3
3 4 5
2 3 6
7 1 2
Matriz não atende Vandermonde

Outro exemplo de execução (com M=3, N=3):

Digite matriz 3x3
1 4 16
1 3 9
1 2 4
Matriz atende Vandermonde totalmente

Outro exemplo de execução (com M=2, N=3):

Digite matriz 2x3
1 4 16
1 7 49
Matriz atende Vandermonde parcialmente

p3_2015_sols

//————————————locacao devolucao de filmes————————————

```
#include<iostream>
#define N 5
using namespace std;
void le_pra(int p[N]){
  int i;
  for (i=0;i<N;i++){
     cin>>p[i];
  }
}
void imp_pra(int p[N]){
  int i;
  for (i=0;i<N;i++){
    cout<<p[i]<<" ";
  }
  cout<<endl;
}
int main() {
  int p[N];
  int fil,qtd;
  char op='x';
  cout<<"Entre a situacao da prateleira: "<<endl;
  le_pra(p);
  imp_pra(p);
  while (op != 'f'){
    cout<<"(a) alugar: (d) devolver ou (f) finalizar ";
    cin>>op;
    if (op=='f') {
        cout<<"Fim da execucao"<<endl;
        break;
    }
    if (op=='a'){
        while (1==1){
            cin>>fil>>qtd;
            if (fil==-1){
                break;
            }
            if (p[fil]>=qtd){
                cout<<"Alugado filme "<<fil<<endl;
                p[fil]=p[fil]-qtd;
            }
            else {
                cout<<"Filme "<<fil<<" sem exemplares"<<endl;
            }
        }
    }
    if (op=='d'){
        while (1==1){
           cin>>fil>>qtd;
           if (fil==-1){
                break;
           }
           p[fil]=p[fil]+qtd;
           cout<<"Devolvido filme "<<fil<<endl;
        }
    }
    cout<<"Situacao da prateleira: "<<endl;
    imp_pra(p);
  }
}
```



```
//-------------------------maior que 2 vizinhos-------------------------
#include<iostream>
#define M 3
#define N 4
using namespace std;
int calcula_2maior(int mat[M][N]){
  int i,j,ve,vd,qtd=0;
  for (i=0;i<M;i++){
    for (j=0;j<N;j++){
        if (j==0){
            ve=0;
        }
        else{
            ve=mat[i][j-1];
        }
        if (j==N){
            vd=0;
        }
        else {
            vd=mat[i][j+1];
        }
        if ((mat[i][j]>ve)&&(mat[i][j]>vd)){
            qtd++;
        }
    }
  }
  return qtd;
}
int main() {
  int mat[M][N]={{7,15,4,11},{5,3,20,0},{1,6,9,8}};
  int xx;
  xx=calcula_2maior(mat);
  cout<<"Qtde de elementos 2-maior: "<<xx<<endl;
}
//---------------------eliminação de elemento no vetor--------------
#include<iostream>
#define N 10
using namespace std;
int imprime_vetor(int vet[N]){
   int i;
   for (i=0;i<N;i++){
    cout<<vet[i]<<" ";
   }
   cout<<endl;
}
int main() {
   int vet[N];
   int i,pos,qual;
   cout<<"Informe vetor: "<<endl;
   for (i=0;i<N;i++){
     cin>>vet[i];
   }
   cout<<"Indique posicao a eliminar: ";
   cin>>pos;
   qual=vet[pos];
   for (i=pos;i<N-1;i++){
      vet[i]=vet[i+1];
   }
   vet[N-1]=0; //99 para testar
   cout<<"Vetor final: ";
   imprime_vetor(vet);
   cout<<"Valor eliminado: "<<qual;
}
```



```
//-----------------------formacao de triangulos-----------------
#include<iostream>
#define M 5
#define N 4
using namespace std;
int triangulo(float a, float b, float c){
   if ((a<b+c)&&(b<a+c)&&(c<a+b)){ return 1; }
   else { return 0; }
}
int main() {
   float mat[M][N]={{3,4,5,8},{0,0,0,0},{1,2,3,4},{4,3,2,3},{1,1,1,3}};
   int sentinela[N];
   int i,j,qual;
   for (i=0;i<M;i++){
        qual=0;
        for(j=2;j<N;j++){
           if (triangulo(mat[i][0], mat[i][1], mat[i][j])){
                sentinela[qual]=j;
                qual++;
           }
        }
        if (qual==0){
            cout<<"Linha "<<i<<" nao permite formar triangulos"<<endl;
        }
        else{
            cout<<"Linha "<<i<<" colunas 0 e 1 com ";
            for (j=0;j<qual;j++){
                cout<<sentinela[j]<<" ";
            }
            cout<<endl;
        }
    }
}
//-----------------------consumo de eletricidade---------------------
#include<iostream>
#define N 10
using namespace std;
int main() {
  float res[N]={20,40,60,80,100,120,140,160,180,200};
  float lim,extr=0,nv;
  int i;
  cout<<"Digite valor limite: "<<endl;
  cin>>lim;
  cout<<"Valores antes do reajuste: "<<endl;
  for (i=0;i<N;i++){
      cout<<res[i]<<" ";
  }
  cout<<endl;
  for (i=0;i<N;i++){
      nv=res[i];
      if ((res[i]>lim)&&(res[i]<=lim*1.4)) {nv=res[i]*1.1;}
      if ((res[i]>lim*1.4)&&(res[i]<=lim*1.7)){ nv=res[i]*1.25;}
      if (res[i]>lim*1.7) {nv=res[i]*1.5;}
      extr=extr+nv-res[i];
      res[i]=nv;
  }
  cout<<"Valores depois do reajuste: "<<endl;
  for (i=0;i<N;i++){
      cout<<res[i]<<" ";
  }
  cout<<endl;
  cout<<"Valor extra arredacadado "<<extr;
}    // furo: 140>100*1.4 deu 1, dai mudei para 1.4000001
```



```
//-------------------------miolo ou borda------------------------------
#include<iostream>
#define M 3
#define N 4
using namespace std;
int miolomolecascadura(int mat[M][N]){
   int i,j,den=0,fora=0;
   for (i=0;i<M;i++){
      for(j=0;j<N;j++){
          if ((i==0)||(i==M)||(j==0)||(j==N)){
              fora=fora+mat[i][j];
          }
          else {
            den=den+mat[i][j];
          }
      }
    }
    if (fora>=den) {return 1;} else {return 0;}
}
int main() {
//  int mat[M][N]={{1,4,2,5},{23,5,4,9},{9,3,7,5}};
  int mat[M][N]={{1,4,1,5},{1,25,14,2},{2,3,7,5}};
  int res;
  res=miolomolecascadura(mat);
  if (res==1){
    cout<<"Borda"<<endl;
  }
  else {
    cout<<"Miolo"<<endl;
  }
}
//-------------------------futebol n times------------------
#include<iostream>
#include<cmath>
#define N 10
using namespace std;
int main(){
//  int vet[N]={-2, 6, 1, 5, -1, 0};
  // time 1 perdeu de 2 gols, time 2=jogou com 6, time 3 jogou com 1,
  // 4 jogou com 5, 5 perdeu de 1 e 6 empatou a 0
  int vet[N]={0,0,-3,9,1,-3,6,2,-1,3};
  int mdif=-99999, qemp=0, onde;
  int i;
  cout<<"Resultado da rodada: "<<endl;
  for (i=0;i<N;i++){
    cout<<vet[i]<<" ";
  }
  cout<<endl<<"--------------------"<<endl;
  for (i=0;i<N;i++){
    if ((vet[i]<0)&&(abs(vet[i])>mdif)){mdif=abs(vet[i]);}

    if (vet[i]>0){
        onde = vet[i]-1;
        vet[i]=abs(vet[onde]);
    }
    if (vet[i]==0){qemp++;}
  }
  cout<<"Saldo de gols: "<<endl;
  for (i=0;i<N;i++){
    cout<<vet[i]<<" ";
  }
  cout<<endl<<"Maior diferenca de gols: "<<mdif<<endl;
  cout<<"Quantidade de empates: "<<qemp/2<<endl;
```



```
}
//------------------------ soma dos menores de cada linha e coluna-------
#include<iostream>
#define M 4
using namespace std;
int smenorColLin(int mat[M][M], int k){
  int i,menorl,menorc;
  menorl=menorc=999999;
  for (i=0;i<M;i++){
     if (mat[i][k]<menorc){
        menorc=mat[i][k];
     }
     if (mat[k][i]<menorl){
        menorl=mat[k][i];
     }
  }
  return menorc+menorl;
}
int main() {
  int mat[M][M]={{1,4,2,5},{23,5,4,9},{9,3,7,5},{6,5,4,2}};
  int j,soma=0;
  for (j=0;j<M;j++){
    soma=soma+smenorColLin(mat,j);
  }
  cout<<"Soma dos menores: "<<soma<<endl;
}
//------------------ divisiveis pelo primeiro valor ------------------
#include<iostream>
#define N 10
using namespace std;
int main() {
  int vet[N]={3,4,2,5,6,3,7,9,12,15};
  int j,soma=0,divisor;
  divisor=vet[0];
  for (j=0;j<N;j++){
    if (vet[j]%divisor==0){
        soma=soma+vet[j];
        vet[j]=-1;
    }
  }
  cout<<"Vetor gerado: "<<endl;
  for (j=0;j<N;j++){
    cout<<vet[j]<<" ";
  }
  cout<<endl<<"Soma: "<<soma<<endl;
}
//--------------------- modelo de Vandermonde --------------------
#include<iostream>
#define M 2 // numero de linhas
#define N 3 // numero de colunas
#include<cmath>
using namespace std;
int avalia_vandermonde(float mat[M][N]){
  int simnao=1,j,k; // comeca com sim}
  for (j=0;j<M;j++){
    for (k=1;k<N;k++){
        if (mat[j][k]!=pow(mat[j][1],k)){simnao=0;}
    }
  }
  if (simnao==0) {return -1;}
  if ((simnao==1)&&(j!=k)){return 0;}
  if ((simnao==1)&&(j==k)){return 1;}
}
```



```
int main(){
//  float mat[M][N]={{3,4,5},{2,3,6},{7,1,2} };  nao atende
//  float mat[M][N]={{1,4,16},{1,3,9},{1,2,4}};  atende totalmente
  float mat[M][N]={{1,4,16},{1,7,49}};  // atende parcialmente
  int z;
  z=avalia_vandermonde(mat);
  if (z==-1){cout<<"Matriz nao atende vandermonde";}
  if (z==0){cout<<"Matriz atende vandermonde parcialmente";}
  if (z==1){cout<<"Matriz atende vandermonde totalmente";}
}
```

