CI208 - Programação de Computadores — Exame Final

Nome: _____ 2º 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)

A sequência de *Fibonacci* é uma sequência de números que começa com 0, 1 e que daí por diante, cada elemento (**e**) terá um valor (**v**) igual à soma do último (**u**) com o penúltimo (**p**) elemento, como demonstrado na tabela 1. Escreva um programa em

elemento	1	2	3	4	5	6	7	···
ultimo	-	-	1	1	2	3	5	···
penúltimo	-	-	0	1	1	2	3	···
valor (u + p)	0	1	1	2	3	5	8	···

Tabela 1: Valores da sequência de *Fibonacci*.

C++ que leia uma sequência de números, e que para cada número *n* lido, calcule a sequência dos *n* primeiros números de Fibonacci. Quando o usuário inserir um número negativo na entrada, o programa deverá parar a sua execução.

Exemplo de execução:

```
Entre com o valor: 1
Sequência: 0
Entre com o valor: 2
Sequência: 0 1
Entre com o valor: 3
Sequência: 0 1 1
Entre com o valor: 4
Sequência: 0 1 1 2
Entre com o valor: 11
Sequência: 0 1 1 2 3 5 8 13 21 34 55
Entre com o valor: -1
Fim de execução.
```

Questão 2 (50 pontos)

Faça um programa em C++ que crie uma matriz simétrica. O programa deve calcular a soma dos valores que estão acima da diagonal principal e a soma dos valores que estão abaixo da diagonal e compará-los. A parte que possuir a maior soma será espelhada na outra parte da matriz. Considere para essa operação que a matriz utilizada será quadrada de dimensões N linhas e N colunas, (N estabelecido via diretiva `#define`).

Neste programa deve ser criada e usada uma função chamada `matriz_simetrica` que recebe como argumento a matriz a ser transformada em simétrica. Essa função deve calcular a soma dos valores acima da diagonal principal, a soma dos valores abaixo da diagonal principal e espelhar a matriz original com os valores da parte que tiver a maior soma na outra parte da matriz, a partir da diagonal principal.

OBS.: Para obter os dados da matriz e para imprimir os dados da matriz, considere que já existem as funções `void ler_matriz (int matriz[][N])` e `void imprimir_matriz(int mat[][N])` respectivamente, já codificadas e disponíveis para uso. Sua solução deve apenas chamar estas funções onde necessário.

Exemplo de execução (para N = 4):

```
Entre com a matriz:
1  2  3  4
5  6  7  8
9  10 11 12
13 14 15 16
Soma acima da diagonal principal: 36
Soma abaixo da diagonal principal: 66
Matriz simétrica:
1 5 9 13
5 6 10 14
9 10 11 15
13 14 15 16
```

CI208 - Programação de Computadores — Exame Final

Nome: _____ 2º 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)

Escrever um programa em C++ que leia do teclado um par de números inteiros não negativos a e b, correspondentes aos limites inferior e superior, nesta ordem, de um intervalo $[a, b]$, com $a \leqslant b$. A seguir, ler ainda do teclado um número inteiro $N(1 \leqslant N \leqslant 5)$ e um conjunto de N números inteiros (que denominaremos divisores). Na seqüência, o programa deverá listar os números do intervalo lido, incluindo seus limites, que não são divisíveis por nenhum dos N números do conjunto de divisores lido. Ao final, mostrar a quantidade de números do intervalo que foram listados. Seu programa deve validar o intervalo informado, o valor de N e também a inexistência de zeros no conjunto de divisores informado.

Exemplo de execução:

```
[a,b]: 25 15
Limites inválidos
```

Outro exemplo de execução:

```
[a,b]: 15 25
Qtde. divisores (N): 8
Qtde. inválida
```

Outro exemplo de execução:

```
[a,b]: 15 25
Qtde. divisores (N): 3
Divisor 1: 2
Divisor 2: 0
    Inválido - reentre
Divisor 2: -3
Divisor 3: 4
Valores em [a,b] não divisíveis
   pelos divisores informados:
      17  19  23  25
Números listados: 4
```

Questão 2 (50 pontos)

Fazer programa em C++ que lê uma matriz quadrada $N \times N$, (N estabelecido via diretiva `#define`), calcula a soma dos valores na diagonal principal e a soma dos valores na diagonal secundária, imprimindo na tela o resultado. O cálculo deve ser feito por uma função que recebe a matriz e retorna os dois resultados.

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 onde necessário.

Exemplo de execução (com N=3):

```
Informe Matriz:
3  4  5
2  3  6
7  1  2
Soma diagonal principal = 8
Soma diagonal secundaria = 15
```

CI208 - Programação de Computadores — Exame Final

Nome: _____ 2º 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)

Um inteiro n é dito um **número perfeito** se ele é igual a soma de todos os seus divisores positivos, excluindo ele mesmo. Por exemplo, 6 é um número perfeito, pois seus divisores são 1, 2 e 3 e 1+2+3 = 6. O número 28 também é perfeito, pois 1+2+4+7+14=28. Faça um programa em C++ que leia uma sequência de números inteiros $n_1, n_2, \ldots, n_k$ terminada por zero e informe ao usuário, para cada n_i lido, se ele é um número perfeito ou não. O número 0 apenas indica o final da sequência e não deve ser considerado para a verificação da propriedade.

Exemplo de execução:

```
Entre com um inteiro: 2
2 não é um número perfeito.
Entre com um inteiro: 28
28 é um número perfeito.
Entre com um inteiro: 79
79 não é um número perfeito.
Entre com um inteiro: 8128
8128 é um número perfeito.
Entre com um inteiro: 0
```

Questão 2 (50 pontos)

Em um engradado de bebidas, pode-se ter: Água (1), refrigerante (2), cerveja (3) ou nada (0). Um refrigerante possui, em média, 10,8 gramas de açúcar para cada 100 ml de bebida, uma cerveja tem em média 5% de álcool na sua composição e a água não possui nada de álcool ou açúcar.

Dado um engradado de bebidas, representado por uma matriz $M \times N$, sendo os valores de M e N estabelecidos pela diretriz `#define`, escreva um programa em C++ que leia a matriz e **utilize uma função** que receba a matriz e calcule: o número total de garrafas, o total de litros de bebida; o total de gramas de açúcar; o total em litros de álcool; e a concentração por litro de açúcar em todo o engradado. Todas as garrafas do engradado possuem a mesma capacidade e ela é fornecida pelo usuário no início do programa.

OBS.: Para a leitura da matriz que representa o engradado, considere que já existe a função `void ler_matriz (int matriz[][N])` já codificada e disponível para uso. Sua solução deve apenas chamar estas funções onde necessário.

Exemplo de execução (para M=2 e N=6):

```
Quantos litros têm as garrafas? 0.6
Entre com o engradado:
1 2 3 3 2 0
0 0 2 2 1 1
Total de garrafas: 9
Total de líquido: 5.4 l
Total de açúcar: 259.2 g
Total de álcool: 0.06 l
Concentração total de açúcar: 48 g/l.
```

Outro exemplo de execução (para M=2 e N=6):

```
Quantos litros têm as garrafas? 1
Entre com o engradado:
3 3 3 3 2 3
0 0 2 2 1 1
Total de garrafas: 10
Total de líquido: 10 l
Total de açúcar: 324 g
Total de álcool: 0.25 l
Concentração total de açúcar: 32.4 g/l.
```

fin_2014s2_sols

```
//----------------- sequencia de fibonacci-----------------
//q03.cpp - sequencia de fibonacci
#include<iostream>
using namespace std;

int main() {
    int i,u,p,v,n;
    while (1==1){
       cout<<"Entre com o valor ";
       cin>>n;
       if (n<0){
          cout<<"Fim da execucao "<<endl;
          return 0;
       }
       p=0;
       u=1;
       if (n==1){
          cout<<"0"<<endl;
       }
       if (n==2){
          cout<<"0 1"<<endl;
       }
       if (n>2){
           cout<<"0 1 ";
           i=2;
          while(i<n){
             v=p+u;
             cout<<v<<' ';
             p=u;
             u=v;
             i++;
             }
             cout<<endl;
       }
    }
}
//------------------cria matriz simetrica----------------
// q04.cpp - cria matriz simetrica
#include<iostream>
#define N 4
using namespace std;

int main() {
   int m[N][N];
   int i,j;
   cout<<"Informe a matriz ";
   for (i=0;i<N;i++){
      for(j=0;j<N;j++){
         cin>>m[i][j];
      }
   }
   int aba,aci;
   aba=0;
   aci=0;
   for (i=0;i<N;i++){
      for(j=0;j<N;j++){
          if (i>j){
             aba=aba+m[i][j];
          }
          if (j>i){
             aci=aci+m[i][j];
```



```
         }
      }
   }
   cout<<"Soma acima "<<aci<<endl;
   cout<<"Soma abaixo "<<aba<<endl;
   if (aci<aba){
      for (i=0;i<N;i++){
         for(j=0;j<N;j++){
            if (i<j){
               m[i][j]=m[j][i];
            }
         }
      }
   }
   else {
      for (i=0;i<N;i++){
         for(j=0;j<N;j++){
            if (j<i){
               m[i][j]=m[j][i];
            }
         }
      }
   }
   for (i=0;i<N;i++){
      for(j=0;j<N;j++){
          cout<<m[i][j]<<' ';
      }
      cout<<endl;
   }
}
//-----------------acha divisores-------------------

//q05.cpp - localiza divisores
#include<iostream>
using namespace std;

int main() {
   int a,b,n,i,qd,listados;
   int divs[5];
   listados=0;
   cout<<"[a,b] ";
   cin>>a>>b;
   if (b<a){
      cout<<"Limites invalidos "<<endl;
      return 0;
      }
   cout<<"Qtde divisores (N): ";
   cin>>n;
   if ((n<1)||(n>5)){
      cout<<"Qtd. invalida ";
      return 0;
      }
   i=0;
   for (i=0;i<n;i++){
      cout<<"Divisor "<<i+1<<": ";
      cin>>divs[i];
      if (divs[i]==0){
         cout<<"Invalido - reentre "<<endl;
         i--;
      }
   }
   cout<<"valores nao divisiveis "<<endl;
   while (a<=b){
```



```
      qd=0;
      for (i=0;i<n;i++){
         if ((a%divs[i])==0){
            qd++;
         }
      }
      if (qd==0){
          cout<<a<<' ';
          listados++;
      }
      a++;
   }
   cout<<endl<<"Numeros listados "<<listados<<endl;
}
//-----------soma dos valores da d. principal e secundaria -------------
#include<iostream>
#define N 3
using namespace std;
int soma2 (int m[N][N], int & ss){
  int i,j,sp=0;
  ss=0;
  for (i=0;i<N;i++){
    for (j=0;j<N;j++){
        if (i==j){
            sp=sp+m[i][j];
        }
        if (i+j==N-1){
            ss=ss+m[i][j];
        }
    }
  }
  return sp;
}
int main(){
  int m[N][N]={{3,4,5},{2,3,6},{7,1,2}};
  int sp,ss;
  sp=soma2(m,ss);
  cout<<"Soma diagonal princial: "<<sp<<endl;
  cout<<"Soma diagonal secundária: "<<ss<<endl;
}
//-----------------numeros perfeitos-------------------------
#include<iostream>
using namespace std;
int main(){
  int num,dii,som;
  while(1==1){
    cout<<"Entre com um inteiro: ";
    cin>>num;
    if (num==0){
        break;
    }
    dii=1+(num/2);
    som=0;
    while(dii>0){
        if (num%dii==0){
            som=som+dii;
        }
        dii--;
    }
    if ((num==som)&&(num!=1)){
        cout<<num<<" e um numero perfeito."<<endl;
    }
    else {
```



```
        cout<<num<<" nao e um numero perfeito."<<endl;
    }
  }
}
//------------------engradado de bebidas-------------------------
#include<iostream>
#define M 2
#define N 6
using namespace std;
int bebida(int m[M][N], float capa, float & tlit, float & tacu, float & talc,
     float & conc, int & totg){
     int i,j;
     totg=0;
     for (i=0;i<M;i++){
        for (j=0;j<N;j++){
            if (m[i][j]!=0){
                tlit=tlit+capa;
                totg++;
            }
            if (m[i][j]==2){
                tacu=tacu+(capa/0.1)*10.8;
            }
            if (m[i][j]==3){
                talc=talc+capa*0.05;
            }
        }
     }
     conc=tacu/tlit;
}
int main(){
  int i,j,totg;
//  int m[M][N]={{1,2,3,3,2,0},{0,0,2,2,1,1}};
  int m[M][N]={{3,3,3,3,2,3},{0,0,2,2,1,1}};
  float capa,tlit,tacu,talc,conc;
  cout<<"Quantos litros tem as garrafas ? ";
  cin>>capa;
  tlit=tacu=talc=conc=0;
  bebida(m,capa,tlit,tacu,talc,conc,totg);
  cout<<"Total de garrafas: "<<totg<<endl;
  cout<<"Total de liquido: "<<tlit<<" l."<<endl;
  cout<<"Total de acucar: "<<tacu<<" g."<<endl;
  cout<<"Total de alcool: "<<talc<<" l."<<endl;
  cout<<"Concentracao total de acucar: "<<conc<<" g/l."<<endl;
}
```

