

Nome: _____ 1º semestre 2017

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 , a função `float fabs(float x)` para calcular o valor absoluto (módulo) de um número real x ,
- 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 ajude um usuário a calcular os juros de um investimento financeiro. Se a conta contém um saldo inferior a R\$ 5.000,00, então o banco paga juros mensais de 2%. Para saldos superiores a R\$ 5.000,00 o banco paga 3% de juros. O programa deve solicitar do usuário o saldo da conta e informar o saldo após a aplicação da taxa. Para fazer o cálculo solicitado, defina e chame no programa principal a função `calinveste()`, que recebe como parâmetro um valor de saldo e retorna o novo saldo.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário. A exibição e leitura de dados devem ser feitas exclusivamente pelo programa principal.

Exemplo de execução:

Digite saldo atual: 12546
Novo saldo: 12922.38

Outro exemplo de execução:

Digite saldo atual: 4000
Novo saldo: 4080.00

Questão 2 (50 pontos)

Considere as seguintes definições:

Definição 1: Um número inteiro n é um quadrado perfeito se existe outro inteiro i tal que $i * i = n$. Ou seja, n pode ser formado pelo quadrado de um número (menor) i .

Definição 2: Um número inteiro n é um cubo perfeito se existe outro inteiro i tal que $i * i * i = n$. Ou seja, n pode ser formado pelo cubo de um número (menor) i .

Usando C++ , definir a função `quadsCubos()` e um programa que utilize a mesma.

A função deve receber como parâmetros dois valores inteiros a e b (estritamente positivos), e deve

produzir dois resultados: *qQuadrados*, representando a quantidade de quadrados perfeitos no intervalo fechado $[a, b]$, e *qCubos*, representando a quantidade de cubos perfeitos no intervalo fechado $[a, b]$.

O seu programa principal deve ler do teclado dois números inteiros positivos (diferentes de 0) $n1$ e $n2$, e exibir o número de quadrados perfeitos e cubos perfeitos no intervalo $[n1, n2]$, valores estes calculados usando a função `quadsCubos()`.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário. A exibição e leitura de dados devem ser feitas exclusivamente pelo programa principal.

Exemplo de execução:

Informe números do intervalo: 2 9
2 quadrados perfeitos nesse intervalo
1 cubos perfeitos nesse intervalo

Outro exemplo de execução:

Informe números do intervalo: 1 9
3 quadrados perfeitos nesse intervalo
2 cubos perfeitos nesse intervalo

Outro exemplo de execução:

Informe números do intervalo: 8 40
4 quadrados perfeitos nesse intervalo
2 cubos perfeitos nesse intervalo

Nome: _____ 1º semestre 2017

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 , a função `float fabs(float x)` para calcular o valor absoluto (módulo) de um número real x ,
- 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 uma função em C++ de nome `compat()` que recebe 3 números naturais e retorna o valor 1 (um) se eles são **compatíveis**, ou 0 (zero) caso contrário). Os números para serem compatíveis precisam ser divisíveis entre si, considerando que o divisor é sempre um número menor que o dividendo. Escreva também o programa em C++ que lê pelo teclado 3 números naturais e usando a função anteriormente especificada, verifique se são compatíveis ou não, imprimindo mensagens de acordo com o resultado.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário. A exibição e leitura de dados devem ser feitas exclusivamente pelo programa principal.

Exemplo de execução:

Digite 3 números naturais: 5 10 20
São compatíveis

Outro exemplo de execução:

Digite 3 números naturais: 24 3 6
São compatíveis

Outro exemplo de execução:

Digite 3 números naturais: 4 3 7
Não são compatíveis

Questão 2 (50 pontos)

Usando C++, definir a função `saqueNotas()` e um programa que utilize a mesma.

O programa principal simula um caixa eletrônico que possui somente notas de R\$ 100 e notas de R\$ 10. O programa deve solicitar que o usuário digite uma sequência de valores de saque e, a cada novo saque, deverá ser impresso quantas notas de R\$ 100 e quantas notas de R\$ 10 deverão ser entregues ao cliente. No início do programa deverá ser informado qual o valor disponível no caixa eletrônico,

que deverá ser reduzido a cada novo saque efetuado com sucesso. Nenhum saque poderá entregar mais que 9 notas de R\$ 10. O programa finaliza quando o caixa estiver sem saldo.

A função `saqueNotas()`, que recebe o saldo s e a quantidade r a ser retirada, deverá calcular e devolver a quantidade de notas de cada tipo de cédula, e deverá retornar também o valor 1 (um) se há saldo em caixa para o saque e 0 (zero) caso contrário.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário. A exibição e leitura de dados devem ser feitas exclusivamente pelo programa principal.

Exemplo de execução:

Saldo em caixa: 1000
Valor do saque: 300
Cédulas: 3 de R\$ 100 e 0 de R\$ 10.
Valor do saque: 700
Cédulas: 7 de R\$ 100 e 0 de R\$ 10.
Sem saldo, caixa desativado.

Outro exemplo de execução:

Saldo em caixa: 1000
Valor do saque: 800
Cédulas: 8 de R\$ 100 e 0 de R\$ 10.
Valor do saque: 210
Saldo insuficiente.
Valor do saque: 190
Cédulas: 1 de R\$ 100 e 9 de R\$ 10.
Valor do saque: 10
Cédulas: 0 de R\$ 100 e 1 de R\$ 10.
Sem saldo, caixa desativado.

Nome: _____ 1º semestre 2017

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 , a função `float fabs(float x)` para calcular o valor absoluto (módulo) de um número real x ,
- 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 data mágica é aquela em que o produto do dia pelo mês tem seus dois últimos dígitos iguais aos dois dígitos finais do ano. Por exemplo, 10 de Junho de 1960 é uma data mágica, pois Junho é o 6º mês e 10×6 é 60, que é igual aos dois dígitos finais do ano.

Faça um programa em C++ que peça ao usuário uma data na forma *dia mes ano* (onde $1 \leq dia \leq 31$, $1 \leq mes \leq 12$, e $1900 \leq ano \leq 2500$) e mostre uma mensagem dizendo se a data informada é ou não mágica, ou se é inválida.

Para verificar se uma data é mágica ou inválida, deverá ser criada e chamada a função de nome `ehMagico()` que recebe como parâmetros três valores inteiros representando respectivamente *dia*, *mes* e *ano*, e que retorna o valor 1 (um) se a data é mágica, ou 0 (zero) caso contrário. Caso o valor de qualquer um dos parâmetros esteja fora dos intervalos indicados acima, a função deve retornar o valor -1 (um negativo).

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário. A exibição e leitura de dados devem ser feitas exclusivamente pelo programa principal.

Exemplo de execução:

Data: 10 6 1960
Data é mágica

Outro exemplo de execução:

Data: 30 6 1960
Data não é mágica

Outro exemplo de execução:

Data: 30 12 1960
Data é mágica

Outro exemplo de execução:

Data: 30 6 1890

Data inválida

Questão 2 (50 pontos)

Escrever um programa em C++ que leia do teclado um conjunto de pares de números. O término da leitura de dados deve ser indicado por um par de números zero. Cada par é composto por um número real r e outro inteiro i , este último não devendo ser nulo. O programa deverá analisar cada par (r, i) e exibir na tela as partes inteira e fracionária de r , exibindo também mensagem indicando se a parte inteira de r é ou não divisível por i .

Ao final do programa, o total tp de pares que atenderam à propriedade verificada e o total geral tg de pares válidos deverão ser mostrados.

Para verificar se a parte inteira de um número real é ou não divisível por um outro número inteiro, deverá ser utilizada a função `analisaNumeros()`, que você deverá desenvolver. Esta função deverá receber como parâmetros um número real nr e um número inteiro div , e devolver como resultados as partes inteira pi e fracionária pf do número real nr , e um inteiro at cujo valor é 0 (zero) se pi não é divisível por div ou 1 (um) caso contrário.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário. A exibição e leitura de dados devem ser feitas exclusivamente pelo programa principal.

Exemplo de execução:

Informe par (real e inteiro): 5.389 3
Parte inteira: 5
Parte fracionária: 0.389
5 não é divisível por 3
Informe par (real e inteiro): 12.2525 6
Parte inteira: 12
Parte fracionária: 0.2525
12 é divisível por 6
Informe par (real e inteiro): 7 0
Entrada inválida
Informe par (real e inteiro): -4.567 2
Parte inteira: -4
Parte fracionária: -0.567
-4 é divisível por 2
Informe par (real e inteiro): -0.5 -2
Parte inteira: 0
Parte fracionária: -0.5
0 é divisível por -2
Informe par (real e inteiro): 15 6
Parte inteira: 15
Parte fracionária: 0
15 não é divisível por 6
Informe par (real e inteiro): 0 0
Pares validos: 5
Pares com a propriedade: 3

p2_2017_sols

```
//-----1-----
// prova:2 tipo:a q:1
#include<iostream>
using namespace std;
float calinveste(float saldo){
    if (saldo<5000){
        return saldo+saldo*0.02;
    }
    else {
        return saldo+saldo*0.03;
    }
}
int main() {
    float sa;
    cout<<"Digite saldo atual ";
    cin>>sa;
    sa = calinveste(sa);
    cout<<"Novo saldo "<<sa<<endl;
}
//----- 2-----
// prova:2 tipo:a q:2
#include<iostream>
#include<cmath>
using namespace std;
int equad(int x){
    int z;
    z=sqrt(x);
    return (z*z)==x;
}

int ecub(int x) {
    int z;
    z=cbrt(x);
    return (z*z*z)==x;
}

void quadsCubos(int a, int b, int & qtq, int & qtc){
    int i;
    qtq=0;
    qtc=0;
    i=a;
    while (i<=b) {
        if (equad(i)){
            qtq++;
        }
        if (ecub(i)){
            qtc++;
        }
        i=i+1;
    }
    return;
}

int main(){
    int k,w,z1,z2;
    cout<<"Informe numeros do intervalo ";
    cin>>k>>w;
    quadsCubos(k,w,z1,z2);
    cout<<z1<<" quadrados perfeitos"<<endl;
    cout<<z2<<" cubos perfeitos"<<endl;
}
```

```

//----- 3 -----
// prova:2 tipo:b q:1
#include<iostream>
using namespace std;
int compat(int a,int b, int c){
    if ((a<=b) && (a<=c) && (b<=c)){
        return ((b%a)==0) && ((c%a)==0) && ((c%b)==0);
    }
    if ((a<=b) && (a<=c) && (c<=b)){
        return ((b%a)==0) && ((c%a)==0) && ((b%c)==0);
    }
    if ((b<=a) && (b<=c) && (a<=c)){
        return ((a%b)==0) && ((c%b)==0) && ((c%a)==0);
    }
    if ((b<=a) && (b<=c) && (c<=a)){
        return ((a%b)==0) && ((c%b)==0) && ((a%c)==0);
    }
    if ((c<=a) && (c<=b) && (a<=b)){
        return ((a%c)==0) && ((b%c)==0) && ((b%a)==0);
    }
    if ((c<=a) && (c<=b) && (b<=a)){
        return ((a%c)==0) && ((b%c)==0) && ((a%b)==0);
    }
}

int main() {
    int x,y,z;
    cout<<"Digite 3 numeros naturais ";
    cin>>x>>y>>z;
    if (compat(x,y,z)){
        cout<<"Sao compativeis";
    }
    else {
        cout<<"Nao sao compativeis";
    }
}

//----- 4 -----
// prova:2 tipo:b q:2
#include<iostream>
using namespace std;
int saqueNotas(float s, float q, int & n100, int & n10 ){
    if (q<=s) {
        n100 = q/100;
        n10=(q-(n100*100))/10;
        return 1;
    }
    else {
        return 0;
    }
}

int main(){
    float saldo,saque;
    int resultado,notas100,notas10;
    cout<<"Saldo em caixa ";
    cin>>saldo;
    while (saldo>0) {
        cout<<"Valor do saque ";
        cin>>saque;
        resultado=saqueNotas(saldo,saque,notas100,notas10);
        if (resultado==1){
            cout<<"Cedulas: "<<notas100<<" de 100 e "<<notas10<<" de 10"<<endl;
        }
    }
}

```

```

    saldo = saldo - saque;
}
else {
    cout<<"Saldo insuficiente "<<endl;
}
}
if (saldo<=0){
    cout<<"Sem saldo, caixa desativado"<<endl;
}
}
// ----- 5 -----
// prova:2 tipo:c q:1
#include<iostream>
using namespace std;
int ehMagico(int dia, int mes, int ano){
    int oba,rabo;
    if (((dia<1)||(dia>31)) || ((mes<1)||(mes>12)) || ((ano<1900)||(ano>2500))) {
        return -1;
    }
    else {
        oba = (dia*mes)%100;
        rabo = ano%100;
        if (rabo==oba){
            return 1;
        }
        else{
            return 0;
        }
    }
}

int main() {
    int d,m,a,dica;
    cout<<"Data: ";
    cin>>d>>m>>a;
    dica = ehMagico(d,m,a);
    if (dica==1){
        cout<<"Data e magica"<<endl;
    }
    else {
        if (dica==0) {
            cout<<"Data nao e magica"<<endl;
        }
        else {
            cout<<"Data invalida "<<endl;
        }
    }
}

//----- 6-----
// prova:2 tipo:c q:2
#include<iostream>
using namespace std;
int analisaNumeros(float nr, int ni, int & pi, float & pf, int & enaoe) {
    pi = nr;
    pf = nr - pi;
    if ((pi%ni)==0) {
        enaoe = 1;
    }
    else {
        enaoe =0;
    }
}
}

```

```

int main() {
    float numeror,partef;
    int numeroi,partei,simnao;
    int tp,tg;
    tp=0;
    tg=0;
    numeror=1;
    numeroi=1;
    while (1==1) {
        cout<<"Informe par (real e inteiro): ";
        cin>>numeror>>numeroi;
        if ((numeroi==0) && (numeror!=0)){
            cout<<"Entrada invalida"<<endl;
        }
        else {
            if ((numeror==0) && (numeroi==0)){
                cout<<"Pares validos "<<tg<<endl;
                cout<<"Pares com a propriedade "<<tp<<endl;
                return 0;
            }
            else {
                tg++;
                analisaNumeros(numeror,numeroi,partei,partef,simnao);
                cout<<"Parte inteira "<<partei<<endl;
                cout<<"Parte fracionaria "<<partef<<endl;
                if ((partei%numeroi)==0) {
                    cout<<partei<<" e divisivel por "<<numeroi<<endl;
                    tp++;
                }
                else {
                    cout<<partei<<" nao e divisivel por "<<numeroi<<endl;
                }
            }
        }
    }
}

```