

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**.

Outro exemplo de execução:

*Digite valores x e n: -3 3
O valor de -3^3 é: -27*

Outro exemplo de execução:

*Digite valores x e n: 3 -3
O expoente deve ser positivo.*

Outro exemplo de execução:

*Digite valores x e n: 3 0
O valor de 3^0 é: 1*

Questão 1 (50 pontos)

Um programa para gerenciar as contas de um banco deve possuir algum mecanismo para calcular o saldo dos clientes após as movimentações diárias. Escreva um programa em C++ que receba o valor da quantia em conta, e após o código da operação realizada e valor associado. O programa deve retornar o saldo em conta. Se o saldo for negativo, ele deve imprimir mensagem informando que o cliente entrou no cheque especial.

As operações permitidas são:

100 depósito em dinheiro

101 depósito em cheque

102 depósito de outra conta

200 saque em dinheiro

201 saque com cheque

202 saque para outra agência

Exemplo de execução:

*Saldo: 2000
202 1000
Saldo em conta: 1000*

Outro exemplo de execução:

*Saldo: 1000
200 1500
Você entrou no Cheque Especial !!*

Questão 2 (50 pontos)

Escreva um programa em C++ que leia dois números inteiros, x , e n , sendo n não-negativo, e que calcule o valor de x^n . Não é permitido o uso da função `float pow(float x, float y)` ou de qualquer outra função pré-existente.

Exemplo de execução:

*Digite valores x e n: 3 4
O valor de 3^4 é: 81*

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 atleta treina em uma pista de corrida circular de 400 metros. Ao final do treino, com base na quantidade de voltas dadas e no tempo total do treino (em horas, minutos e segundos), ele quer saber:

- O percurso total do treino (em quilômetros);
- O tempo médio por quilômetro (em horas, minutos e segundos);
- O tipo de atleta, de acordo com o tempo médio por quilômetro:
 - Atleta de competição, se o tempo médio está no intervalo entre 0 e 4 minutos (inclusive);
 - Atleta participante, se o tempo médio está no intervalo acima de 4 e abaixo de 7 minutos (7 inclusive);
 - Participante de caminhada, se o tempo médio é acima de 7 minutos.

Faça um programa em C++ que auxilie o atleta a ter essas respostas. Para o tempo, se a sua solução tiver valores abaixo de segundos, ignore a parte decimal/fracionária após os segundos.

Exemplo de execução

No. voltas: 13
Tempo treino (hh mm ss): 0 55 23
Percurso total (km): 5.2
Tempo médio/km (hh mm ss): 0 10 39
Tipo: Participante de caminhada

Outro exemplo de execução

No. voltas: 13
Tempo treino (hh mm ss): 0 20 30
Percurso total (km): 5.2
Tempo médio/km (hh mm ss): 0 3 57
Tipo: Atleta de competição

Questão 2 (50 pontos)

Escreva um programa em C++ que leia do teclado conjuntos de três medidas a , b , e c (reais e positivas) correspondentes aos lados de triângulos. Para cada tripla lida, o programa deverá verificar e informar se é ou não realmente possível a montagem de um triângulo. Sendo possível, o perímetro do triângulo deverá ser calculado e mostrado no monitor de vídeo. Não sendo possível, calcular e mostrar a dimensão mínima ACIMA DA QUAL a adição desta a algum dos lados menores informados possibilitará a formação de um triângulo. O conjunto de triplas se encerra com a informação da primeira medida nula ou negativa; esta tripla não faz parte do conjunto de dados a processar. Ao final do programa deverão ser mostrados o total de triângulos possíveis de montagem, seu perímetro médio, o total de triângulos não possíveis de montagem, a média dos incrementos faltantes e o total de triplas lidas. **Observação:** seu programa não precisa validar se as medidas informadas são reais e positivas. Lembre-se que para formar um triângulo, a soma de dois lados quaisquer é sempre maior que a medida do terceiro lado, isto é, dados a , b , e c , eles formam um triângulo se e somente se $a + b > c$, $b + c > a$, $a + c > b$

Exemplo de execução:

Medidas: 3 4 5
Formam um triângulo. Perímetro: 12
Medidas: 10 2 4
Não formam um triângulo.
Incremento mínimo > 4
Medidas: 2.5 3 4.5
Formam um triângulo. Perímetro: 10
Medidas: 6 6 5
Formam um triângulo. Perímetro: 17
Medidas: 1 1 7
Não formam um triângulo.
Incremento mínimo > 5
Medidas: -3 1 2
Triângulos montados: 3
Perímetro médio: 13
Triângulos não montados: 2
Média dos incrementos faltantes: 4.5
Total de triplas: 5

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)

Faça um programa em C++ que leia 4 números inteiros H_a, M_a, H_d e M_d , onde $H_a : M_a$ representa o horário atual (H_a horas e M_a minutos) e $H_d : M_d$ representa o horário programado em um despertador (H_d horas e M_d minutos). Seu programa deve calcular e mostrar quanto tempo resta até que o alarme desperte (quantas horas e quantos minutos). Se os horários forem **idênticos**, seu programa deve indicar que o alarme *está despertando*, e, se o horário programado for **anterior** ao horário atual, seu programa deve indicar que o alarme *já despertou*. Considere que os valores H_a e H_d estão sempre entre 0 e 23 e que os valores M_a e M_d estão sempre entre 0 e 59. Além disso, considere que o horário programado no despertador é referente ao **dia atual** e nunca ao dia seguinte.

Exemplo de execução:

```
10 12
14 15
4 horas e 3 minutos para despertar
```

Outro exemplo de execução:

```
16 34
22 06
5 horas e 32 minutos para despertar
```

Outro exemplo de execução:

```
23 42
03 06
Alarme já despertou!
```

Outro exemplo de execução:

```
09 42
09 42
Alarme está despertando!
```

Outro exemplo de execução:

```
13 45
```

13 40

Alarme já despertou!

Questão 2 (50 pontos)

Faça um programa em C++ que leia um conjunto de valores inteiros n_1, n_2, \dots, n_d representando as faces obtidas em d lançamentos de um dado de seis faces e imprime uma mensagem indicando se o dado é *regular* ou se está *viciado*. Considere que todo valor lido n_i está entre 1 e 6, isto é, $1 \leq n_i \leq 6$, e que o final do conjunto de valores é indicado quando o usuário digitar o valor 0 (zero), que **não** deve ser contado como um lançamento do dado. Além disso, considere que o dado está viciado se o número de ocorrências de alguma de suas faces for **maior que o dobro** do valor esperado, definido pelo número total de lançamentos multiplicado pela probabilidade de ocorrência da respectiva face. Isto é,

Dado viciado \iff Ocorrências de alguma das faces $>$

$$2 \times \text{Número total de lançamentos} \times \frac{1}{6}$$

Exemplo de execução:

```
1
1
2
5
4
6
0
Dado regular
```

Outro exemplo de execução:

```
6
4
3
4
1
4
4
2
0
Dado viciado
```