

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;
- 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)

Sabe-se que 4 segmentos podem formar um quadrilátero apenas se a medida do maior segmento for menor que soma das medidas dos outros 3. Escreva um programa em C++ que recebe do usuário as medidas de 4 segmentos, verifica sua validade (valores reais positivos e não nulos), avalia a possibilidade de formar algum quadrilátero com as medidas informadas e, caso possível, calcula e mostra o perímetro do polígono obtido. Para a verificação da possibilidade de formar o quadrilátero, deve ser definida e usada a função `analisa4()`, que recebe como entrada as medidas de 4 segmentos e devolve ao programa que a chamou: -1, caso alguma das medidas informadas seja inválida; 0, caso as medidas sejam válidas mas não permitam a formação de um quadrilátero; OU o perímetro do quadrilátero, caso as medidas sejam válidas e possibilitem a formação do mesmo.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

Medidas (4 segmentos): 10 2 -5 6
Medidas inválidas

Outro exemplo de execução:

Medidas (4 segmentos): 10 2 0.5 6
Medidas não formam um quadrilátero

Outro exemplo de execução:

Medidas (4 segmentos): 10 2 5 6
Perímetro do quadrilátero formado: 23

Questão 2 (50 pontos)

Codifique um programa em C++ para auxiliar um leiloeiro a identificar os três maiores lances em suas atividades de vendas. O programa deverá receber vários lances para compra de um produto em leilão, onde cada lance é caracterizado por um código do comprador (inteiro, positivo e diferente de 0), e por um valor do lance, em reais. Quando um lance possuir 0 no código do comprador ou no valor da oferta, significa que o leilão acabou e o programa deverá mostrar os três maiores lances, em ordem decrescente, e encerrar o processamento. Para identificar os três maiores lances defina e utilize a função `Seleciona_Melhores_Ofertas()`

que recebe cada lance (com código do comprador e valor do lance) juntamente com as variáveis para armazenar os três melhores lances, cada um composto do código do comprador e do valor de seu lance. Caso o valor do lance seja menor ou igual ao maior valor de lance existente, a função deverá retornar 0 (significando que o lance é inválido), e em caso contrário deverá retornar +1. Caso a função retorne o valor 0, o programa deverá dar mensagem de "Lance inválido: Ignorado.", e em caso contrário deverá dar mensagem de "Lance OK: Processado."

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

Digite o Lance 1: 4 1012.50
Lance OK: Processado.
Digite o Lance 2: 1 980
Lance inválido: Ignorado.
Digite o Lance 2: 3 1050
Lance OK: Processado.
Digite o Lance 3: 4 1200.99
Lance OK: Processado.
Digite o Lance 4: 5 1250
Lance OK: Processado.
Digite o Lance 5: 3 1300
Lance OK: Processado.
Digite o Lance 6: 0 0
Foram processados 5 Lances.
1o. melhor Lance: 1300 do comprador 3
2o. melhor Lance: 1250 do comprador 5
3o. melhor Lance: 1200.99 do comprador 4

Outro exemplo de execução:

Digite o Lance 1: 4 1012.50
Lance OK: Processado.
Digite o Lance 2: 3 1050
Lance OK: Processado.
Digite o Lance 3: 0 0
Foram processados 2 Lances.
1o. melhor Lance: 1050 do comprador 3
2o. melhor Lance: 1012.50 do comprador 4

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;
- 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 loja tem um plano de desconto progressivo para os seus clientes, onde quanto mais eles compram, mais desconto eles ganham. Além disso, a loja tem um plano de fidelidade que garante um desconto alternativo ainda maior aos clientes mais assíduos, de acordo com a tabela abaixo:

Valor da compra	Desconto Normal	Desconto Fidelidade
Até R\$ 200,00	0%	2%
de 200,01 a 300,00	5%	7%
de 300,01 a 550,00	10%	13%
A partir de 550,01	15%	19%

Escreva um programa em C++ que peça ao usuário o valor de uma compra e o tipo do cliente que a efetuou. EM seguida o programa deve exibir o valor dado como desconto e o valor que foi efetivamente pago pelo cliente. Para calcular o desconto, deve ser definida e usada a função chamada `calc_desc()`, que recebe o valor total da compra do cliente e um valor que indique se o cliente tem plano de fidelidade (0 se não tiver, 1 se tiver) e que retorne o valor em reais do desconto aplicado ao valor da compra.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

Valor da compra e tipo de cliente:
1072.30 1
Total de desconto: 203.73
Total a pagar: 868.56

Outro exemplo de execução:

Valor da compra e tipo de cliente:
247.37 0
Total de desconto: 12.36
Total a pagar: 235.01

Questão 2 (50 pontos)

Fulano, Beltrano e Cicrano estão disputando quem tem mais sorte. Para isso, eles pegaram um dado de seis lados e, um de cada vez, lançou o dado e anotou o número obtido. Eles fizeram diversas jogadas e agora querem descobrir quem foi o vencedor de cada rodada. Faça um programa completo

em C++ que os auxilie a descobrir quantas rodadas cada um venceu. Obtenha do teclado o número de rodadas que foram realizadas e crie uma função chamada `vencedor_rodada()` que recebe como parâmetros os números que cada um tirou em uma determinada rodada. Faça sua função retornar 1, 2 ou 3 caso Fulano, Beltrano ou Cicrano tenha vencido aquela rodada, nessa ordem. Ao final do seu programa, mostre quantas rodadas cada jogador venceu.

Obs. 1: Assuma que houve apenas um vencedor, sem empates durante as rodadas.

OBS. 2: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

Digite numero de rodadas do jogo: 3
Digite numeros obtidos pelos jogadores:
1 2 3
4 5 6
1 3 6
Fulano venceu 0 vezes
Beltrano venceu 0 vezes
Cicrano venceu 3 vezes

Outro exemplo de execução:

Digite numero de rodadas do jogo: 3
Digite numeros obtidos pelos jogadores:
6 2 1
1 2 4
4 1 5
Fulano venceu 1 vezes
Beltrano venceu 0 vezes
Cicrano venceu 2 vezes

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;
- 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 verifique se a quantidade em estoque de um produto de uma loja acabou ou não toda vez que houve uma retirada. O programa deve pedir ao usuário a quantidade inicial do produto, e depois solicitar do usuário a quantidade vendida do produto, informando quando não houver quantidade suficiente em estoque para a venda. O programa termina mostrando a quantidade final do estoque. O programa deve usar a função `estoque()` para verificar se há quantidade suficiente e para atualizar a quantidade do estoque. Ela deve receber como parâmetros a quantidade em estoque e a quantidade vendida, e deve devolver o novo valor do estoque, que deve permanecer o mesmo caso o estoque não consiga cobrir a quantidade vendida.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

```
Qtde. inicial do estoque: 20
Qtde. vendida: 12
Qtde. em estoque: 8
```

Outro exemplo execução:

```
Qtde. inicial do estoque: 13
Qtde. vendida: 20
Estoque insuficiente: 13
```

Questão 2 (50 pontos)

O *pôquer curitibano* é uma variante do jogo na qual cada jogador começa com duas cartas e, a cada rodada, pode escolher trocar uma de suas cartas por uma carta comprada do baralho. Nessa variante, apenas as cartas 2, 3, ..., 9, 10 e A (Ás) são utilizadas, sendo que a pontuação de cada carta é igual ao seu valor, exceto a carta A que tem pontuação 14. Ganha o jogador que, após k rodadas, tenha na mão o par de maior valor ou as cartas de maiores valores (caso nenhum jogador tenha um par).

Faça um programa em C++ que auxilie um jogador a tomar as decisões mais lógicas numa partida de pôquer curitibano. Crie uma função chamada `troca_carta()` que recebe três parâmetros: as duas cartas que o jogador tem em mãos e a carta que foi comprada do monte nessa rodada. Essa função deve verificar se é vantajoso trocar alguma

carta da mão pela comprada e retornar as cartas trocadas, se for o caso, e um inteiro representando se houve troca (1) ou não (0). No programa principal, leia o número k de rodadas da partida e dois inteiros representando as cartas que o jogador tem na mão no início do jogo (14 representa a carta A). Para cada uma das k rodadas, leia um inteiro representando a carta comprada. Utilizando a função `troca_carta()` seu programa deve obter, em cada rodada, uma mão melhor que a anterior, se possível. Caso haja troca de cartas a mensagem "apostar!" deve ser impressa. Caso contrário, a mensagem "mesa." é que deve ser mostrada. Se o jogador terminar a última rodada com um par de ases na mão, a mensagem "apostar tudo!!!" deve ser impressa, independente da rodada na qual o par foi obtido.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

```
Número de rodadas: 3
Mão inicial: 2 7
Rodada 1: 3
Mão: 3 7 -- apostar!
Rodada 2: 3
Mão: 3 3 -- apostar!
Rodada 3: 14
Mão: 3 3 -- mesa.
```

Outro exemplo de execução:

```
Número de rodadas: 3
Mão inicial: 14 3
Rodada 1: 2
Mão: A 3 -- mesa.
Rodada 2: 14
Mão: A A -- apostar!
Rodada 3: 10
Mão: A A -- apostar tudo!!!
```

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;
- 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 receba do usuário a medida do raio de uma circunferência e uma escolha: 1 se o usuário deseja saber o perímetro da circunferência, 2 se o usuário deseja saber a área do círculo correspondente. O programa deverá usar uma função de nome `calcCirculo()` que recebe o raio e a escolha e retorna o valor do cálculo realizado. A entrada e saída de dados deve ser completamente manipulada pelo programa principal.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

```
Informe raio da circunf.: 3
Deseja perimetro (1) ou área (2)?: 1
Perímetro = 18.8496
```

Outro exemplo de execução:

```
Informe raio da circunf.: 3
Deseja perimetro (1) ou área (2)?: 2
Área = 28.2743
```

Questão 2 (50 pontos)

Daenerys e Drogo entendiados inventaram um jogo de adivinhação. Neste jogo o oponente 1 inventa uma senha que consiste de um número de 3 dígitos não repetidos entre 1 e 4. São exemplos de senhas VÁLIDAS: 123, 231 e 412; As senhas 212 e 315 são senhas INVÁLIDAS. O oponente 2 deverá dar palpites para adivinhar a senha. Para cada palpite feito devem ser dadas as seguintes dicas: (a) quantos dígitos do palpite aparecem em qualquer posição na senha; (b) quantos dígitos do palpite aparecem na posição correta na senha (vide Exemplo de execução abaixo).

Sua missão é implementar este jogo no computador. Para isso você deve escrever uma função em C++ de nome `dicas()` que recebe como parâmetros uma *senha* e um *palpite*. A função deve devolver a quantidade de dígitos do *palpite* que aparecem em *senha* (`acertdig`) e a quantidade de dígitos do *palpite* que aparecem em *senha* na mesma posição (`acertdigpos`). Além disso, a função deve retornar o valor 0 (zero) se o palpite for incorreto, ou 1 (um) se o palpite estiver correto. Escreva também o programa principal que lê a senha do oponente 1 e depois lê os palpites do oponente 2

e, usando os resultados da função `dicas()`, imprime as dicas correspondentes até que este acerte a senha. O programa somente termina quando o oponente 2 adivinha a senha.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

```
Oponente 1, entre sua senha: 341
Oponente 2, adivinhe a senha: 324
DICAS
Qtd. dígitos corretos: 2
Qtd. dígitos corretos na pos. correta: 1
Oponente 2, adivinhe a senha: 321
DICAS
Qtd. dígitos corretos: 2
Qtd. dígitos corretos na pos. correta: 2
Oponente 2, adivinhe a senha: 421
DICAS
Qtd. dígitos corretos: 2
Qtd. dígitos corretos na pos. correta: 1
Oponente 2, adivinhe a senha: 341
Você acertou. Parabéns!
```

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;
- 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 obtenha do usuário 4 valores inteiros e imprima na tela mensagem indicando se a soma dos dois maiores números é divisível pelo ou divisor do produto dos 3 menores valores informados. Para testar esta relação entre os valores deverá ser criada e usada a função `ehDivisor()`, que recebe 4 valores inteiros como parâmetros, e retorna o valor 1 (um) se a relação “soma divisível pelo produto” se verifica, 2 (dois) se a relação “soma divisor do produto” se verifica, ou 0 (zero) caso contrário.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

Indique 4 números inteiros: 1 7 3 2
Propriedade não se verifica.

Outro exemplo de execução do programa:

Indique 4 números inteiros: 2 3 4 8
12 é divisor de 24.

Outro exemplo de execução do programa:

Indique 4 números inteiros: 20 1 2 4
24 é divisível por 8.

(zero) caso contrário.

OBS.: A função NÃO DEVE mostrar dados na tela ou solicitar valores do usuário.

Exemplo de execução:

Carga horaria: 90
Informe agendamentos:
1
Consulta agendada.
1
Consulta agendada.
2
Consulta agendada.
1
Consulta nao agendada.
2
Consulta agendada.
Total de consultas agendadas: 4

Outro exemplo de execução:

Carga horaria: 50
Informe agendamentos:
2
Consulta agendada.
1
Consulta agendada.
Total de consultas agendadas: 2

Questão 2 (50 pontos)

Faça um programa em C++ para realizar o agendamento de consultas ambulatoriais do SUS. Dois tipos de consultas são realizadas: (1) consulta normal e (2) consulta de retorno. Uma consulta normal possui 30 minutos de duração, enquanto uma consulta de retorno requer 15 minutos. Primeiro, o usuário deve informar a carga horária do médico em minutos. Depois, os agendamentos são realizados informando o tipo da consulta para cada paciente enquanto há carga horária suficiente para mais um agendamento. Ao final, o programa deve informar o número total de consultas agendadas.

Para resolver o problema proposto pede-se o desenvolvimento e uso de uma função de nome `agendamento()` que recebe como entrada o tipo de consulta e a carga horária disponível, e devolve esta carga horária atualizada de acordo com o tempo da consulta, devolvendo também o valor 1 (um) caso a consulta tenha sido agendada ou 0

p2_2015_sols

```
#include<iostream>
using namespace std;
// ----- 1-----
int analisa4(float a, float b, float c, float d){
    if((a<=0) || (b<=0) || (c<=0) || (d<=0)){return -1;}
    if((a>(b+c+d)) || (b>(a+c+d)) || (c>(a+b+d)) || (d>(a+b+c))){return 0;}
    return a+b+c+d;
}
int q1(){
    float a,b,c,d,p,z;
    cout<<"Medidas (4 segmentos): ";
    cin>>a>>b>>c>>d;
    z=analisa4(a,b,c,d);
    if(z==-1){cout<<"Medidas invalidas"<<endl;}
    if(z==0){cout<<"Medidas nao formam quadrilatero"<<endl;}
    if(z>0){cout<<"Perimetro do quadrilatero formado "<<z<<endl;}
}

//----- 2 -----
int seleciona_melhores_ofertas(int c, float l, int & c0, float & l0,
                                int & c1, float & l1, int & c2, float & l2){
    if (l<=l0) {return 0;}
    if ((l>l0) && (l>l1) && (l>l2)){l2=l1; l1=l0; l0=l; c2=c1; c1=c0; c0=c; return 1;}
    if ((l>l1) && (l>l2)){l2=l1; l1=l; c2=c1; c1=c; return 1;}
    if (l>l2){l2=l; c2=c; return 1;}
}
int q2(){
    int lance=1,r;
    int c,c0,c1,c2;
    float l,l0,l1,l2;
    l0=l1=l2=0;
    c=l=1;
    while (l==1){
        cout<<"digite o lance "<<lance<<" ";
        cin>>c>>l;
        if ((c==0) || (l==0)){
            break;
        }
        r=seleciona_melhores_ofertas(c,l,c0,l0,c1,l1,c2,l2);
        if (r==0){
            cout<<"Lance invalido, ignorado"<<endl;
        }
        if (r==1){
            cout<<"Lance OK: processado"<<endl;
            lance++;
        }
    }
    cout<<"Foram processados "<<lance<<" lances."<<endl;
    if (l0>0){
        cout<<"1. melhor lance> "<<l0<<" do comprador "<<c0<<endl;
    }
    if (l1>0){
        cout<<"2. melhor lance> "<<l1<<" do comprador "<<c1<<endl;
    }
    if (l2>0){
        cout<<"3. melhor lance> "<<l2<<" do comprador "<<c2<<endl;
    }
}
//----- 3 -----
float calc_desco(float co, int pf){
```

```

    float d=0;
    if ((co<=200)&&(pf==1)){d=0.02;}
    if ((co>200)&&(co<=300)&&(pf==0)){d=0.05;}
    if ((co>200)&&(co<=300)&&(pf==1)){d=0.07;}
    if ((co>300)&&(co<=550)&&(pf==0)){d=0.1;}
    if ((co>300)&&(co<=550)&&(pf==1)){d=0.13;}
    if ((co>550)&&(pf==0)){d=0.15;}
    if ((co>550)&&(pf==1)){d=0.19;}
    return co*d;
}
int q3(){
    int tc;
    float r,vc;
    cout<<"Valor da compra e tipo cliente"<<endl;
    cin>>vc>>tc;
    r=calc_desco(vc,tc);
    cout<<"Total do desconto: "<<r<<endl;
    cout<<"Valor a pagar: "<<vc-r;
}

// ----- 4 -----
int vencedor_rodada(int f, int b, int c){
    if ((f>b)&&(f>c)){return 1;}
    if ((b>f)&&(b>c)){return 2;}
    if ((c>f)&&(c>b)){return 3;}
}
int q4(){
    int tf=0, tb=0, tc=0;
    int z,f,b,c,i,nr;
    cout<<"Digite numero de rodadas do jogo: ";
    cin>>nr;
    cout<<"Digite numeros obtidos pelos jogadores: "<<endl;
    for (i=0;i<nr;i++){
        cin>>f>>b>>c;
        z=vencedor_rodada(f,b,c);
        if (z==1){tf++;}
        if (z==2){tb++;}
        if (z==3){tc++;}
    }
    cout<<"Fulano venceu "<<tf<<" vezes"<<endl;
    cout<<"Beltrano venceu "<<tb<<" vezes"<<endl;
    cout<<"Cicrano venceu "<<tc<<" vezes"<<endl;
}
// -----5 -----
int estoque(int & es, int qv){
    if (qv<es) {
        es=es-qv; return 1;
    }
    else{
        return 0;
    }
}
int q5(){
    int est, v,r ;
    cout<<"Qtde inicial do estoque ";
    cin>>est;
    cout<<"Qtde vendida: ";
    cin>>v;
    r=estoque(est, v);
    if (r==0){cout<<"Estoque insuficiente: "<<est<<endl;}
    if (r==1){cout<<"Qtde. em estoque: "<<est<<endl;}
}

```

```

}
// -----6 -----
int troca_carta(int & c1, int & c2, int cc){
    if (c1==c2){return 0;}
    if ((c1==cc)&&(c2!=cc)){c2=cc; return 1;}
    if ((c2==cc)&&(c1!=cc)){c1=cc; return 1;}
    if ((cc>c1)&&(c2>c1)){c1=cc; return 1;}
    if ((cc>c2)&&(c1>c2)){c2=cc; return 1;}
    return 0;
}
int q6(){ // temos problema com a conversao de 14 em A...
    int i,nr,a,b,carta,z;
    cout<<"Numero de rodadas: ";
    cin>>nr;
    cout<<"Mao inicial: ";
    cin>>a>>b;
    for (i=1;i<=nr;i++){
        cout<<"Rodada "<<i<<" ";
        cin>>carta;
        cout<<endl;
        z=troca_carta(a,b,carta);
        if ((i==nr)&&(a==14)&&(b==14)){cout<<"A A -- apostar tudo!!!"<<endl;}
        if (z==1){cout<<a<<" "<<b<<" -- apostar!"<<endl;}
        if (z==0){cout<<a<<" "<<b<<" -- mesa."<<endl;}
    }
}

//-----7 -----
float calcCirculo(float r, int tipo){
    if (tipo==1){return 2*3.14159265*r;}
    if (tipo==2){return 3.14159265*r*r;}
}
int q7(){
    float ra;
    int ti;
    cout<<"Informe o raio da circunf: ";
    cin>>ra;
    cout<<"Deseja perimetro (1) ou area (2) ?: ";
    cin>>ti;
    if (ti==1){cout<<"Perimetro: "<<calcCirculo(ra,ti)<<endl;}
    if (ti==2){cout<<"Area: "<<calcCirculo(ra,ti)<<endl;}
}
//-----8 -----
int dicas(int senha, int palpi, int & adig, int & apos) {
    int s1,s2,s3,p1,p2,p3,dc=0,pc=0,saux,paux;
    saux=senha;
    paux=palpi;
    s1 = saux%10;
    saux=saux/10;
    s2 = saux%10;
    saux=saux/10;
    s3=saux;
    p1 = paux%10;
    paux = paux/10;
    p2=paux%10;
    paux=paux/10;
    p3=paux;
    if((p1==s1)|| (p1==s2)|| (p1==s3)){dc++;}
    if((p2==s1)|| (p2==s2)|| (p2==s3)){dc++;}
    if((p3==s1)|| (p3==s2)|| (p3==s3)){dc++;}
    if(p1==s1){pc++;}
    if(p2==s2){pc++;}
}

```

```

    if(p3==s3){pc++;}
    adig=dc;
    apos=pc;
    if (senha==palpi) {return 1;} else {return 0;}
}
int q8(){
    int ss,pp,qa,qp,z;
    qa=0;
    qp=0;
    cout<<"Oponente 1, entre a sua senha: ";
    cin>>ss;
    while(1==1){
        cout<<"Oponente 2, adivinhe a senha: ";
        cin>>pp;
        z=dicas(ss,pp,qa,qp);
        if (z==1){
            cout<<"Voce acertou, parabens!"<<endl;
            break;
        }
        cout<<"DICAS"<<endl;
        cout<<"Qtd digitos corretos: "<<qa<<endl;
        cout<<"Qtd na posicao correta: "<<qp<<endl;
    }
}

//----- 9 -----
int ehdivisor(int a, int b, int c, int d, int & s, int & p){
    if (((a+b)>(a+c))&&((a+b)>(a+d))&&((a+b)>(b+c))&&((a+b)>(b+d))&&((a+b)>(c+d))){s=a+b;}
    if (((a+c)>(a+b))&&((a+c)>(a+d))&&((a+c)>(b+c))&&((a+c)>(b+d))&&((a+c)>(c+d))){s=a+c;}
    if (((a+d)>(a+c))&&((a+d)>(a+b))&&((a+d)>(b+c))&&((a+d)>(b+d))&&((a+d)>(c+d))){s=a+d;}
    if (((b+c)>(a+c))&&((b+c)>(a+d))&&((b+c)>(a+b))&&((b+c)>(b+d))&&((b+c)>(c+d))){s=b+c;}
    if (((b+d)>(a+c))&&((b+d)>(a+d))&&((b+d)>(b+c))&&((b+d)>(a+b))&&((b+d)>(c+d))){s=b+d;}
    if (((c+d)>(a+c))&&((c+d)>(a+d))&&((c+d)>(b+c))&&((c+d)>(b+d))&&((c+d)>(a+b))){s=c+d;}
    if (((a*b*c)<(a*b*d))&&((a*b*c)<(a*c*d))&&((a*b*c)<(b*c*d))){p=a*b*c;}
    if (((a*b*d)<(a*b*c))&&((a*b*d)<(a*c*d))&&((a*b*d)<(b*c*d))){p=a*b*d;}
    if (((a*c*d)<(a*b*d))&&((a*c*d)<(a*b*c))&&((a*c*d)<(b*c*d))){p=a*c*d;}
    if (((b*c*d)<(a*b*d))&&((b*c*d)<(a*c*d))&&((b*c*d)<(a*b*c))){p=b*c*d;}
    if ((s%p)==0) {return 1;}
    if ((p%s)==0) {return 2;}
    return 0;
}
int q9(){
    int a,b,c,d,s,p,x;
    cout<<"Indique 4 numeros inteiros: ";
    cin>>a>>b>>c>>d;
    x=ehdivisor(a,b,c,d,s,p);
    if (x==1){cout<<s<<" e divisivel por "<<p<<endl;}
    if (x==2){cout<<s<<" e divisor de "<<p<<endl;}
    if (x==0){cout<<"propriedade nao se verifica"<<endl;}
}
//----- 10 -----
int agendamento(int & ch, int co){
    if ((co==1)&&(ch>=30)){ ch=ch-30; return 1;}
    if ((co==2)&&(ch>=15)){ ch=ch-15; return 1;}
    return 0;
}
int q10(){
    int r,a,t=0;
    int carga;
    cout<<"Carga Horaria ";
    cin>>carga;
    while (1==1){

```

```

    cout<<"Informe agendamentos: ";
    cin>>a;
    r=agendamento(carga,a);
    if (r==1){
        cout<<"Consulta agendada."<<endl;
        t++;
    }
    if (r==0){
        cout<<"Consulta nao agendada"<<endl;
    }
    if (carga<15){
        cout<<"Total de consultas agendadas: "<<t<<endl;
        break;
    }
}
}

```

```

int main(){
    q1();
    q2();
    q3();
    q4();
    q5();
    q6();
    q7();
    q8();
    q9();
    q10();
}

```