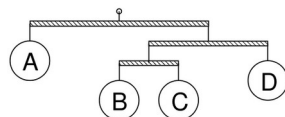


Exercícios de Revisão – Programação 1 – COM06842 – 2022/1

Professor: Jacson Rodrigues Correia da Silva

Fonte dos problemas: inspirados nos problemas da *Olimpíada Brasileira de Informática*

1. O móbile na sala da Zuleide é composto de três hastes exatamente como na figura abaixo:



Para que ele esteja completamente equilibrado, os pesos das quatro bolas A, B, C e D têm que satisfazer todas as seguintes três condições:

$$A = B + C + D; e$$

$$B + C = D; e$$

$$B = C$$

Seu trabalho é ajudar a Zuleide com um programa que receba o valor de cada uma das quatro bolas e que responda se o móbile está equilibrado ou não.

Exemplo de entrada:	Saída referente:	Exemplo de entrada:	Saída referente:
12 3 3 6	Está equilibrado.	82 12 3 2	Não está equilibrado.

Para isso, faça:

- a) Um algoritmo para resolver esse problema.
- b) O código fonte do programa referente ao seu algoritmo.

2. Paula e Carlos vão ganhar **N** bolinhas de chocolate. Como sabem que vão comer tudo muito rápido inventaram uma brincadeira e suas regras:

- Eles vão comer de forma alternada, um depois o outro, sendo que sempre a Paula começa.
- Quem comer a última bolinha ganha a brincadeira.
- A cada vez, cada um só pode comer **M** bolinhas, sendo **M** o valor obtido por um dado, uma única vez, no início da brincadeira.

Você deve escrever um programa que receba o valor **N** e **M** e que seja capaz de: mostrar o histórico da brincadeira; e dizer quem ganhou.

Exemplo de entrada:	Saída referente:
18 5	Total: 18 Quem Joga - Total Paula - 13 Carlos - 8 Paula - 3 Carlos - 0 Carlos Ganhou.

Para isso, faça:

- a) Um algoritmo para resolver esse problema.
- b) O código fonte do programa referente ao seu algoritmo.

Observação: obrigatório utilizar utilizar instrução/ões de iteração, isto é, for, while, do...while.

3. Em uma certa escola, costuma-se praticar uma corrida de 5 voltas ao redor da do prédio da própria escola. Nesse ano, as competições serão sempre entre dois estudantes: estudante1 e estudante2. Os professores vão registrar o tempo de cada volta e você deve criar um programa capaz de ler os tempos registrados de cada um, dizer seu total e o vencedor(a) da corrida.

Exemplo de entrada:	Saída referente:
10 12 14 16 10 11 11 12 14 12	Tempo do estudante 1: 62 Tempo do estudante 2: 60 Vencedor(a): estudante 2

Para isso, faça:

- Um algoritmo para resolver esse problema.
- O código fonte do programa referente ao seu algoritmo.

Observação: obrigatório utilizar utilizar instrução/ões de iteração, isto é, for, while, do...while.

4. Em uma escola antiga, havia o costume de fazer uma rifa invertida. Essa rifa funcionava da seguinte maneira:

- Seu Firmino informava o número da sorte **S** (entre 1 a 500).
- Então, os números de 1 à 500 eram colocados em uma sacola.
- Formava-se uma fila e cada estudante podia colocar a mão na sacola e resgatar um número.
- Quem tirasse o número da sorte **S**, ganhava a rifa invertida.

Esse ano, como Seu Firmino já está velho e com a memória fraca, ele quer usar um programa para lhe ajudar na tarefa de memorizar o número da sorte e informar quando alguém ganhar a rifa.

Assim, Seu Firmino vai informar ao programa o número da sorte **S** e, depois, vai informar cada número que foi resgatado da sacola.

O programa deve armazenar o número da sorte **S** e, no momento que alguém resgatar esse número, deve informar que houve ganhador.

Exemplo de entrada:	Saída referente:
67 28 19 87 501 34 67	Número da sorte: 67 Próximo Próximo Próximo Número inválido Próximo Ganhou!!

Para isso, faça:

- Um algoritmo para resolver esse problema.
- O código fonte do programa referente ao seu algoritmo.

Observações:

- obrigatório utilizar utilizar instrução/ões de iteração, isto é, for, while, do...while.
- deve-se ignorar números inválidos.

5. Crie um programa que peça um número primo ao usuário. Ele deve permanecer em *loop* até que o usuário digite um número que seja realmente um número primo.

6. Crie um programa que imprima o fatorial de um número **N**, informado pelo usuário, na tela.

7. Crie um programa que peça um número ao usuário entre 10 e 99 e que: garanta que o usuário digite um número válido; utilize *switch* para informar qual é a dezena referente ao número.

8. Crie um menu que permita o usuário escolher entre 3 mensagens ocultas. Ao selecionar um número de 1 a 3 (seu programa deve evitar escolhas erradas), utilize *switch* para escrever a mensagem referente: (1) bom dia; (2) você é legal; (3) confio em você.

9. Apresente, com justificativas, o erro (lógico ou de implementação) de cada código abaixo:

a)

```
while (num < 10)
{
    printf("Número: %d\n", num);
}
```

b)

```
if (num == 10);
    printf("o número é igual a 10\n");
```

c)

```
char num;
scanf("Digite um número: %d", num);
```

d)

```
int num;
scanf("%d", &num);
if (num < 10 || num > 0)
    printf("O número está entre 0 e 10\n");
```

e)

```
int i;
for (i=100; i > 10; i++)
    printf("Valor de i: %d\n", i);
```

f)

```
float num;
scanf("%d", &num);
switch(num)
{
    case 1: printf("acertou!!");
    default: printf("errou!!");
}
```

10. Apresente o teste de mesa para cada código abaixo:

a)

```
num = 5;
while (num > 0)
{
    printf("Num: %d\n", num);
    num--;
}
```

b)

```
num = 0;
while (num < 10)
{
    printf("Num: %d\n", num);
    num+=2;
}
```

c)

```
for (pos=3; pos<20; pos+=3)
    if (pos%7==0)
        printf("Divisível por 3 e 7: ", pos);
```