

# Lista de Exercícios

## Leitura de Código

### Questão 01 [1xp]

Qual a saída produzida pelo programa abaixo quando n for, 5, 10 e 11, respectivamente:

```
#include <stdio.h>
void funcao( int a, int b )
{
    float x, y;
    a = 2 * a + 1;
    x = a;
    y = x / 2;
    b = x / 2;
    x = a / 2;
    printf( "1: %i %i %f %f\n", a, b, x, y );
}
int main( void )
{
    int n, a, b;
    scanf( "%i", &n );
    a = n / 10;
    b = n % 10;
    printf( "2: %i %i \n", a, b );
    funcao( a, b );
    printf( "3: %i %i \n", a, b );
    return 0;
}
```

### Questão 02 [2xp]

Qual a saída produzida para o código abaixo? O que a função faz?

```
#include <stdio.h>
#include <math.h>

int misterio( float x1, float x2 )
{
    return ( fabs( x1 - x2 ) < 1e-3 );
}

int main( void )
{
    float x1, x2;
    scanf( "%f %f", &x1, &x2 );
    printf( "%i\n", misterio( x1, x2 ) );
    printf( "%i\n", x1 == x2 );
    return 0;
}
```

### Questão 03 [2xp]

Qual a saída produzida para o código abaixo? O que a função faz?

```
#include <stdio.h>
```

```

#include <math.h>

double misterio( double x1, double y1, double x2, double y2 )
{
    double pot1 = pow( x1 - x2, 2. );
    double pot2 = pow( y1 - y2, 2. );
    return sqrt( pot1 + pot2 );
}

int main( void )
{
    double x1, y1, x2, y2;
    scanf( "%lf %lf", &x1, &y1 );
    scanf( "%lf %lf", &x2, &y2 );
    printf( "%lf", misterio( x1, y1, x2, y2 ) );
    return 0;
}

```

#### Questão 04 [1xp]

Qual a saída produzida pelo programa abaixo?

```

#include <stdio.h>

int main(){
    int i = 3, j = 5;
    int *p = &i, *q = &j;

    printf("%d\n", p == &i);
    printf("%d\n", *p - *q);
    printf("%d\n", **&p);

    return 0;
}

```

#### Questão 05 [2xp]

Qual a saída produzida pelo programa abaixo?

```

#include <stdio.h>

int main( void )
{
    int a, b, *ptr1, *ptr2;
    ptr1 = &a;
    ptr2 = &b;
    a = 1;
    (*ptr1)++;
    b = a + *ptr1;
    *ptr2 = *ptr1 * *ptr2;
    printf( "a = %d, b = %d, *ptr1 = %d, *ptr2 = %d", a, b, *ptr1, *ptr2 );
    return 0;
}

```

#### Questão 06 [1xp]

Qual a saída produzida pelo programa abaixo?

```
#include <stdio.h>

int main( void )
{
    int x = 100, *p, **pp;
    p = &x;
    pp = &p;
    printf( "%i", **pp );
    return 0;
}
```

### Questão 07 [1xp]

Qual a saída produzida pelo programa abaixo?

```
#include <stdio.h>

int func(int a, int* b){
    ++*b;
    return a + *b;
}

int main(){
    int a, b, c;
    c = 10;
    a = 5;
    b = 1;

    a = func(a, &c);
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

### Questão 08 [2xp]

Qual a saída produzida pelo programa abaixo? O que ele faz?

```
#include <stdio.h>

void swap(int* a, int *b){
    int tmp;

    tmp = *a;
    *a = *b;
    *b = tmp;
}

int main(){
    int a, b;

    scanf("%d %d", &a, &b);
    swap(&a, &b);
    printf("%d %d", a, b);
}
```

```
    return 0;
}
```

### Questão 09 [2xp]

O que a saída do código abaixo? Por que isso acontece?

```
#include <stdio.h>
```

```
void swap(int* a, int *b){
    int* tmp;

    tmp = a;
    a = b;
    b = tmp;
}
```

```
int main(){
    int a, b;

    scanf("%d %d", &a, &b);
    swap(&a, &b);
    printf("%d %d", a, b);

    return 0;
}
```

## Ache o Erro

### Questão 01 [2xp]

Qual o erro do programa abaixo?

```
#include <stdio.h>
```

```
int funcao( int *a )
{
    int *b = a;
    int c = *b;
    return c;
}
```

```
int main( void )
{
    int a = 10;
    int *b = &a;
    int c = funcao( &b );
    printf( "%i", c );
    return 0;
}
```

### Questão 02 [1xp]

Qual o erro do programa abaixo?

```
#include <stdio.h>

int main(){
    int x;
    int* pt;
    pt = &x;

    scanf("%d", pt);
    printf("Cuidado com os Red Harrings!")

    return 0;
}
```

### Questão 03 [2xp]

O programa abaixo deveria ler um numero e imprimir ele somado a 5. Qual o erro do programa?

```
#include <stdio.h>

void soma_5(int* x){
    int i;
    for(i=0; i<5; i++, x++);
}

int main(){
    int x;

    scanf("%d", &x);
    soma_5(&x);
    printf("%d", x);

    return 0;
}
```

### Complete o Código

Comandos numerados (**comando1**, **comando2**, ...) devem ser respondidos com apenas um comando, enquanto **comandos** podem ser preenchidos com uma sequência de comandos.

### Questão 01 [2xp]

Complete os comandos abaixo de forma que o programa leia N números inteiros e imprima a soma e o produto dos N números.

```
#include <stdio.h>

void soma_produto(int n, int* soma, int* produto){
    comandos;
}

int main(){
    int n, numero, soma, produto;
    soma = produto = 0;
```

```

scanf("%d", &n);

for(comando1){
    scanf("%d", &numero);
    comando2;
}

printf("%d %d", soma, produto);

return 0;
}

```

### Questão 02 [1xp]

Complete a função de forma que o programa abaixo simule uma transferência de dados, passando o valor da variável origem para a variável destino, um byte por vez.

```
#include <stdio.h>
```

```

void transfere_bytes(int n, int* origem, int* destino){
    comandos;
}

int main(){
    int x, y;
    double w, z;

    scanf("%d %lf", &x, &w);

    transfere_bytes(sizeof(int), &x, &y);
    transfere_bytes(sizeof(double), &w, &z);

    printf("%d %lf\n", y, z);

    return 0;
}

```

## Resolução de Problemas

### Questão 01 [3xp]

Os professores do nPiti estão com um problema em seus equipamentos novos: eles não vieram com mecanismos de segurança implementados! Dessa forma, se a pressão, temperatura ou energia do equipamento não ficarem em valores normais, um grande acidente pode acontecer com os alunos que o manusearem. Para resolver esse problema, eles chamaram os alunos de ITP!

Escreva um programa para monitorar o status de funcionamento do equipamento. O equipamento possui três fatores principais que podem levar a um mal funcionamento: pressão nas válvulas, temperatura do equipamento e nível de energia existente para operar. Os valores normais dessas variáveis são:

Pressão: 0 - 100  
Temperatura: 0 - 70  
Energia: 0 - 100

Existem cinco operações que podem ser feitas sobre o equipamento pelos alunos, que afetam essas variáveis:

**1 - Inserção de materiais:** aumenta a pressão da máquina em 20, a temperatura em 10 e reduz a energia em 15

**2 - Processamento de materiais:** reduz a pressão da máquina em 5, a energia em 25 e a temperatura aumenta em 15

**3 - Refrigeração do equipamento:** reduz a pressão em 7, a temperatura em 10 e aumenta a energia em 5

**4 - Liberação das válvulas:** reduz a pressão em 15, a temperatura em 5 e aumenta a energia em 5

**5 - Overclock do processamento:** aumenta pressão em 15, temperatura em 10 e reduz energia em 20

O seu programa lerá o status inicial da máquina (pressão, temperatura e energia), e em seguida lerá uma sequência de operações sobre a máquina. A cada operação, ela deve imprimir um relatório contendo os três valores das variáveis. Além disso, caso algum dos valores esteja em uma situação limite, o programa deve imprimir uma mensagem, de acordo com a tabela abaixo:

Condição	Mensagem
pressão > 70	Pressão Elevada!
temperatura > 50	Temperatura Elevada!
Energia < 30	Energia Baixa!

Caso algum dos valores atinja uma condição crítica, o programa deve imprimir uma mensagem e encerrar a execução. As mensagens e condições estão listadas na tabela abaixo:

Condição	Mensagem
Pressão >= 100	CABUUUM!
Temperatura >=70	Mossoró Time!
Energia <= 0	Queda de Energia!

#### Entrada

10 20 100  
1

#### Saída

Pressão = 30  
Temperatura = 30

2	Energia = 85
5	
4	Pressão = 25
1	Temperatura = 45
2	Energia = 60
5	
3	Pressão = 40
5	Temperatura = 55
	Energia = 40
	Temperatura Elevada!
	Pressão = 25
	Temperatura = 50
	Energia = 45
	Pressão = 45
	Temperatura = 60
	Energia = 30
	Temperatura Elevada!
	Pressão = 40
	Temperatura = 75
	Energia = 5
	Mossoró Time!

### Questão 02 [1xp] - URI 1161 (Seção Matemática)

Leia dois valores inteiros M e N indefinidamente. A cada leitura, calcule e escreva a soma dos fatoriais de cada um dos valores lidos. Utilize uma variável apropriada, pois cálculo pode resultar em um valor com mais de 15 dígitos.

O arquivo de entrada contém vários casos de teste. Cada caso contém dois números inteiros M ( $0 \leq M \leq 20$ ) e N ( $0 \leq N \leq 20$ ). O fim da entrada é determinado por EOF.

Para cada caso de teste de entrada, seu programa deve imprimir uma única linha, contendo um número que é a soma de ambos os fatoriais (de M e N).

Entrada	Saída
4 4	48
0 0	2
0 2	3

### Questão 03 [1xp] - URI 1198 (Seção Matemática)



Hashmat é um guerreiro cujo seu grupo de soldados se move de um local a outro para lutar contra os seus oponentes. Antes da luta ele apenas calcula uma coisa. A diferença entre a quantidade de soldados que possui e a quantidade de soldados oponentes. A partir desta diferença ele decide se vai ou não lutar. Às vezes Hashmat tem mais soldados do que o seu oponente, mas na maioria das vezes não.

A entrada contém dois números inteiros em cada linha. Estes dois números denotam respectivamente a quantidade de soldados do exército de Hashmat e do seu oponente. Nenhum número de entrada é maior do que  $2^{32}$ . A entrada termina com fim de arquivo (EOF).

Para cada linha de entrada imprima a diferença entre o número de soldados de Hashmat e do seu oponente. Cada saída deve ser impressa em uma linha separada.

Entrada	Saída
10 12	2
10 14	4
100 200	100

#### Questão 04 [5xp]

A Academia da Confederação Galática deseja atualizar o seu programa de simulação de combates, e resolveu contratar os alunos de ITP! De acordo com o capitão da Enterprise, Jean-Luc Picard, a simulação deve seguir as seguintes regras:

- O combate ocorrerá entre duas naves apenas: a Enterprise, identificada pela letra 'E' e uma nave inimiga Klingon, identificada pela letra 'K'
- Cada nave possui os seguintes atributos: **armadura** (quanto ela aguenta de dano antes de explodir), **escudo refletor** (um escudo especial que absorve ataques mais fracos), **mísseis** (tipo de arma, munição limitada), **canhões** (tipo de arma, munição limitada)
- O combate se dá por turnos, de forma que dois ataques não ocorrem ao mesmo tempo. Uma linha de ataque será no formato **Nave Arma**, onde Nave é a letra que identifica a nave que realiza o ataque. Arma é a letra que identifica o tipo de ataque utilizado: 'C' para canhão e 'M' para mísseis.
- As naves sempre se acertam. Quando o ataque for do tipo Míssel, o escudo refletor é ignorado, e a nave é atingida diretamente. Quando o ataque for do tipo Canhão, o ataque é absorvido pelo escudo refletor até que ele seja destruído (valor fique igual a zero), e a partir daí os ataques atingem diretamente a nave adversária. Ataques de Míssel fazem um dano de **20 pontos**. Ataques de Canhão fazem um dano de **10 pontos**. Cada ataque executado reduz a munição existente em uma unidade.
- A simulação ocorrerá por um número N de rodadas, informado ao início do programa.

- Quando a armadura de uma das naves chegar a 0, o programa imprime o vencedor (“**Enterprise Vence!**” ou “**Klingons vencem!**”). Caso as N rodadas terminem e nenhuma das naves seja destruída, o programa deve imprimir “**Empate!**”.

A entrada do programa começa com uma linha contendo o número de ataques N que serão simulados. As duas próximas linhas conterão 4 inteiros cada, sendo respectivamente os atributos da Enterprise (armadura, escudo, munição canhão, mísseis) e da nave Klingon (armadura, escudo, munição canhão, mísseis). A seguir virão N linhas contendo os ataques na forma de dois caracteres: o nome do atacante (**E** ou **K**) e a arma utilizada (**C** ou **M**). Seu programa deve realizar as simulações até que o combate seja vencido por um dos lados ou o número de rodadas termine.

#### Entrada

```
10
50 20 50 3
70 20 50 3
E M
K C
E M
E M
K M
E M
K M
E C
E C
E C
```

#### Saída

```
Sem munição!
Enterprise Vence!
```