Vinicius Henrique de Oliveira Franzote RA: 221024174

**2 –**

#include <stdio.h>

int main() {

int N; // Número de partidas realizadas

int teste = 1; // Contador de testes

while (1) {

// Leitura do número de partidas

scanf("%d", &N);

if (N == 0)

break; // Fim da entrada, encerra o programa

int saldoAtual = 0, saldoMax = 0;

int inicio = 1, fim = 1, melhorInicio = 1, melhorFim = 1;

for (int i = 1; i <= N; i++) {

int golsFavor, golsContra;

scanf("%d %d", &golsFavor, &golsContra);

saldoAtual += (golsFavor - golsContra);

if (saldoAtual > saldoMax) {

saldoMax = saldoAtual;

melhorInicio = inicio;

melhorFim = i;

}

if (saldoAtual < 0) {

saldoAtual = 0;

inicio = i + 1;

}

}

// Impressão do resultado

printf("Teste %d\n", teste++);

if (saldoMax > 0) {

printf("%d %d\n", melhorInicio, melhorFim);

} else {

printf("nenhum\n");

}

printf("\n");

}

return 0;

}

**4 –**

#include <stdio.h>

int main() {

int teste = 1; // Contador de testes

while (1) {

int X1, Y1, X2, Y2; // Coordenadas do retângulo que delimita a fazenda

scanf("%d %d %d %d", &X1, &Y1, &X2, &Y2);

if (X1 == 0 && Y1 == 0 && X2 == 0 && Y2 == 0)

break; // Fim da entrada, encerra o programa

int N; // Número de meteoritos

scanf("%d", &N);

int meteoritos = 0; // Contador de meteoritos dentro da fazenda

for (int i = 0; i < N; i++) {

int X, Y; // Coordenadas do meteorito

scanf("%d %d", &X, &Y);

if (X >= X1 && X <= X2 && Y >= Y2 && Y <= Y1)

meteoritos++;

}

// Impressão do resultado

printf("Teste %d\n", teste++);

printf("%d\n\n", meteoritos);

}

return 0;

}

**6 –**

#include <stdio.h>

int main() {

int teste = 1; // Contador de testes

while (1) {

int N; // Número de depósitos

scanf("%d", &N);

if (N == 0)

break; // Fim da entrada, encerra o programa

// Processamento de cada conjunto de teste

printf("Teste %d\n", teste++);

for (int i = 0; i < N; i++) {

int J, Z; // Valores depositados no cofrinho de João e Zezinho

scanf("%d %d", &J, &Z);

printf("%d\n", J - Z);

}

printf("\n");

}

return 0;

}

**8 –**

#include <stdio.h>

int main() {

double A, G, Ra, Rg; // Preço do álcool, preço da gasolina, rendimento do álcool e rendimento da gasolina

// Leitura dos valores de entrada

scanf("%lf %lf %lf %lf", &A, &G, &Ra, &Rg);

// Cálculo do custo por quilômetro para o álcool e para a gasolina

double custoAlcool = A / Ra;

double custoGasolina = G / Rg;

// Verificação do combustível mais econômico

char combustivelMaisEconomico;

if (custoAlcool < custoGasolina)

combustivelMaisEconomico = 'A';

else

combustivelMaisEconomico = 'G';

// Impressão do resultado

printf("%c\n", combustivelMaisEconomico);

return 0;

}

**10 –**

#include <stdio.h>

int main() {

int N, M; // Número de linhas e colunas do parque

int X, Y; // Coordenadas do microfone

int K; // Número de quadrados percorridos

int minutosCaptados = 0; // Contador de minutos captados pelo microfone

// Leitura dos valores de entrada

scanf("%d %d", &N, &M);

scanf("%d %d", &X, &Y);

scanf("%d", &K);

// Verificação dos minutos de conversa captados

for (int i = 0; i < K; i++) {

int movimento;

scanf("%d", &movimento);

// Verificação se o movimento passa pelo microfone

if ((movimento == 1 && X > 0) ||

(movimento == 2 && X < N) ||

(movimento == 3 && Y < M) ||

(movimento == 4 && Y > 0)) {

minutosCaptados++;

}

}

// Impressão do resultado

printf("%d\n", minutosCaptados);

return 0;

}

**12 –**

#include <stdio.h>

int main() {

int N, P; // Número de competidores e pontuação mínima para ser convidado

int convidados = 0; // Contador de competidores convidados

// Leitura dos valores de entrada

scanf("%d %d", &N, &P);

// Verificação dos competidores convidados

for (int i = 0; i < N; i++) {

int fase1, fase2;

scanf("%d %d", &fase1, &fase2);

if (fase1 + fase2 >= P) {

convidados++;

}

}

// Impressão do resultado

printf("%d\n", convidados);

return 0;

}

**14 –**

#include <stdio.h>

int main() {

int N; // Número de redes

int totalArea = 0; // Área total da região coberta pelas redes

// Leitura do número de redes

scanf("%d", &N);

// Cálculo da área total

while (N--) {

int Xi, Xf, Yi, Yf;

scanf("%d %d %d %d", &Xi, &Xf, &Yi, &Yf);

// Cálculo da área da região coberta pela rede atual e soma à área total

totalArea += (Xf - Xi + 1) \* (Yf - Yi + 1);

}

// Impressão da área total

printf("%d\n", totalArea);

return 0;

}

**16 –**

#include <stdio.h>

int main() {

int A, B, C; // Dimensões dos contêineres

int X, Y, Z; // Dimensões do navio

// Leitura das dimensões dos contêineres e do navio

scanf("%d %d %d", &A, &B, &C);

scanf("%d %d %d", &X, &Y, &Z);

// Cálculo da quantidade máxima de contêineres

int maxContainers = 0;

for (int i = 1; i <= X / A; i++) {

for (int j = 1; j <= Y / B; j++) {

for (int k = 1; k <= Z / C; k++) {

maxContainers++;

}

}

}

// Impressão do resultado

printf("%d\n", maxContainers);

return 0;

}

**18 –**

#include <stdio.h>

int main() {

int A, B, C; // Dimensões dos contêineres

int X, Y, Z; // Dimensões do navio

// Leitura das dimensões dos contêineres e do navio

scanf("%d %d %d", &A, &B, &C);

scanf("%d %d %d", &X, &Y, &Z);

// Cálculo da quantidade máxima de contêineres

int maxContainers = 0;

for (int i = 1; i <= X / A; i++) {

for (int j = 1; j <= Y / B; j++) {

for (int k = 1; k <= Z / C; k++) {

maxContainers++;

}

}

}

// Impressão do resultado

printf("%d\n", maxContainers);

return 0;

}