

Gabarito da Atividade prática - A2

Professores: Amilton C. Rattmann, MSc; Leonardo Gomes, Dr; Vinicius Pozzobon Borin, MSc.

Exercício 1: Faça um algoritmo em linguagem C que recebe como entrada dois valores reais (X e Y) e um símbolo de operação matemática via teclado do usuário (+, -, / ou *). O programa deverá retornar o resultado da operação recebida sobre estes dois números. Deve ser permitido que o usuário faça cálculos com valores negativos e positivos. Também, tome cuidado com a possibilidade de uma divisão por zero. Caso o usuário tente dividir por zero, seu programa deve tratar este problema de alguma maneira. Imprima na tela um teste do seu programa utilizando os dois últimos valores de seu RU da UNINTER como os dois valores de entrada, cada valor do seu RU será um dos valores de entrada X e Y. Por exemplo, se seu RU for 1234567, faça X = 6, Y = 7.

```
#include<stdio.h>
#include<stdlib.h>
int main() {
    //Ler dados
    printf("Digite 2 valores e um operador ( + - / * )\n");
    float x, y, resultado;
    char operador;
    scanf_s("%f %f %c", &x, &y, &operador);
    //Verificar operador e aplicar operador
    switch (operador)
    {
        case ('+'):
            resultado = x + y;
            break;
        case ('-'):
            resultado = x - y;
            break;
        case ('*'):
            resultado = x * y;
            break;

        case ('/'):
            if (y == 0) {
                printf("Nao eh possivel dividir por zero");
                system("pause");
                return 1; //encerra o programa imediatamente
            }
            resultado = x / y;
            break;
        default:
            printf("Operador invalido");
            system("pause");
            return 2; //encerra o programa imediatamente caso um
            operador invalido seja digitado
    }
    //Imprimir resultado
    printf("%.2f %c %.2f = %.2f\n", x, operador, y, resultado);
    system("pause");
    return 0;
}
```

Exercício 2: Você está organizando um bolão de futebol. Segundo suas regras, os apostadores informam o placar do jogo e ganham 10 pontos se acertarem o vencedor ou se acertarem que foi empate, e ganham mais 5 pontos para o placar de cada time que acertarem. A tabela a seguir dá um exemplo, considerando que o placar real de um jogo foi 3x2.

Placar apostado	Pontos	Razão
0x1	0	Não acertou o vencedor e nem o número de gols dos times.
0x2	5	Não acertou o vencedor.
3x5	5	Não acertou o vencedor, mas acertou o número de gols do primeiro time.
1x0	10	Acertou o vencedor, mas não acertou o número de gols dos times.
3x1	15	Acertou o vencedor e o número de gols do primeiro time.
3x2	20	Acertou o vencedor e o número de gols de ambos os times.

Faça um algoritmo em linguagem C que leia como dado de entrada o placar real de um jogo e 5 apostas de jogadores para aquele jogo. Para cada aposta, deve-se ler o nome do apostador e o valor que ele apostou. Para cada aposta, calcule os pontos de cada jogador. Ao final do programa, liste na tela o nome de cada apostador e quantos pontos cada um obteve. Imprima na tela um teste do seu programa utilizando os dois últimos valores de seu RU da UNINTER como o placar real do jogo. Por exemplo, se seu RU for 1234567, faça com que o placar real do jogo seja 6x7.

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

#define APOSTADORES 5
int main() {
    //declaracao de variaveis
    int timeA[5], timeB[5]; //apostas do time A do time B
    char nomes[5][20]; //cinco nomes com ate 20 caracteres cada
    nome.

    int pontos; //variavel que marca os pontos de cada apostador
    int a, b; //placar verdadeiro do time A e time B
    int vitoria; //armazena o vencedor verdadeiro
    int vitoria_aposta; //armazena vencedor da aposta
    //Ler dados

    printf("Digite o placar verdadeiro\n");
    scanf_s("%d %d", &a, &b);

    printf("Digite as apostas seguida do nome do apostador\n");
    for (int i = 0; i < APOSTADORES; i++){
        scanf_s("%d %d", &timeA[i], &timeB[i]);
        gets_s(nomes[i]);
    }

    //Calculo da pontuacao
    //vitoria = 1 time A venceu, -1 time B venceu, 0 empate
    if (a > b) {
        vitoria = 1;
    }
    else if (a < b) {
```

```

        vitoria = -1;
    }
    else {
        vitoria = 0;
    }
    for (int i = 0; i < APOSTADORES; i++){
        pontos = 0;
        //calculo de vitoria da aposta
        if (timeA[i] > timeB[i]) {
            vitoria_aposta = 1;
        }
        else if (a < b) {
            vitoria_aposta = -1;
        }
        else {
            vitoria_aposta = 0;
        }

        //somatorio de pontos
        if (a == timeA[i]) {
            pontos = pontos + 5;
        }
        if (b == timeB[i]) {
            pontos = pontos + 5;
        }
        if (vitoria == vitoria_aposta) { //acertou o vitorioso
            pontos = pontos + 10;
        }

        //imprimir resposta
        printf("Apostador %s fez %d pontos\n", nomes[i],
pontos);

    }

    system("pause");
    return 0;
}

```

Exercício 3: Fazer um algoritmo em linguagem C no qual o usuário vai digitando sucessivamente valores positivos e salvando estes valores em um vetor de dados. Quando o usuário entrar com um valor negativo, o programa deve parar de pedir valores e calcular a média dos valores já fornecidos (excluindo o valor negativo digitado). Para calcular a média, crie uma função que recebe como parâmetro o vetor de dados e retorna para a função principal a média. Imprima a média na tela usando a função principal.

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

float tirar_media(float * vetor, int tam) {

    float somatorio = 0, media; //somatorio recebera a soma de
    todos os valores do vetor

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

        somatorio = somatorio + vetor[i];

    }

    media = somatorio / tam;

    return media;
}

int main() {
    //declaracao de variaveis
    float media, vetor[500]; //maximo de 500 posicoes no vetor
    int tam = 0; //variavel tam marca o tamanho do vetor
    float val; //variavel val marca o ultimo valor lido

    //Ler dados
    printf("Digite quantos valores desejar e finalize com um valor
negativo\n");
    scanf_s("%f", &val); //leitura do primeiro valor fora do loop
    para o caso do usuario começar digitando um valor negativo
    while (val>=0) {
        vetor[tam] = val;
        tam++; // tam = tam+1;
        scanf_s("%f", &val); //leitura do proximo valor
    }

    if (tam == 0) {
        printf("Programa encerrado sem valores digitados");
        system("pause");
        return 1;
    }
    media = tirar_media(vetor, tam);

    //Imprimir resultado
    printf("A media dos valores eh: %.2f\n",media); //imprimindo com
    duas casas decimais

    system("pause");
    return 0;
}
```

Exercício 4: Faça um algoritmo em linguagem C com as seguintes funcionalidades:

- Receba uma matriz como dado de entrada. A dimensão da matriz deve ser de 3x5 (3 linhas e 5 colunas);
- Imprimir a matriz na tela, de uma forma legível;
- Encontre a matriz transposta e imprima-a na tela, de forma legível;
- Imprima na tela novamente a matriz transposta, mas agora multiplicando somente a diagonal principal pelos últimos dois números do seu RU da UNINTER;

```
#include <stdio.h>
#include <stdlib.h>
#define NL 3 // numero de linhas
#define NC 5 // numero de colunas
#define RU 99 // ultimos 2 digitos do RU

int main()
{
    float m[NL][NC];
    float t[NC][NL];

    //Leitura da matriz
    printf("\nEntre com a matriz m[3,5], pressione <Enter> depois de cada valor!\n");
    for (int i = 0; i < NL; i++){
        for (int j = 0; j < NC; j++){
            printf("m[%d, %d] = ", i, j);
            scanf_s("%f", &m[i][j]);
        }
    }
    //imprimir Matriz
    printf("Matriz original:\n");
    for (int i = 0; i < NL; i++){
        for (int j = 0; j < NC; j++){
            printf("%.2f\t", m[i][j]);
        }
        printf("\n"); //quebra de linha
    }
    printf("=====\n");
    //transposta
    for (int i = 0; i < NL; i++) {
        for (int j = 0; j < NC; j++) {
            t[j][i] = m[i][j];
        }
    }
    //Imprimir transposta
    printf("Matriz transposta:\n");
    for (int i = 0; i < NC; i++) {
        for (int j = 0; j < NL; j++) {
            printf("%.2f\t", t[i][j]);
        }
        printf("\n"); //quebra de linha
    }
    printf("=====\n");
    //multiplicando diagonal principal pelo RU;
```

```

    for (int i = 0; i < NL; i++) {
        m[i][i] = m[i][i]*RU;
    }
    //Imprimir transposta
    printf("Matriz transposta multiplicada pelo RU:\n");
    for (int i = 0; i < NC; i++) {
        for (int j = 0; j < NL; j++) {
            printf("%.2f\t", t[i][j]);
        }
        printf("\n");//quebra de linha
    }
    printf("=====\n");
    system("pause");
    return 0;
}

```