



Universidade Federal do Maranhão  
Departamento de Informática  
Ciência da Computação  
Linguagem de programação I  
Prof.: Francisco Glaubos

## Prova 1 - Tipo 2

2023.2

- 1) (5,0 pts) Implemente uma função recursiva em C que calcule e retorne a soma de todos os elementos de um array de inteiros. A função deve receber o array ( $0 < |\text{array}| \leq 100$ ) e o seu tamanho. A função deve ser implementada sem utilizar estruturas de repetição (*loops*) e sem utilizar variáveis globais ou estáticas. A assinatura da função deve ser:

```
int findsum(int a[], int n);
```

Exemplos:

```
int arr[] = {-1, -2, -3, -4, -5};  
Soma: -15
```

```
int arr[] = {-1, 2, -3, 4, 0};  
Soma: 2
```

- 2) (5,0 pts) Crie uma função em C que receba três argumentos: um array `precos[i]` que representa o preço de uma determinada ação no  $i$ -ésimo dia, um array de `bonus[i]` que representa um bônus de

venda que você recebe no  $i$ -ésimo dia além do preço da ação, e um inteiro  $n : n \leq 100$  que representa o número de dias.

Você deseja maximizar seu lucro escolhendo um único dia para comprar uma ação e escolher um dia diferente no futuro para vender essa ação. Retorne o lucro máximo que você pode obter com esta transação. Se você não conseguir nenhum lucro, retorne 0. Você deve implementar a seguinte função, e imprimir na tela o dia da compra e o dia da venda

Obs.: Utilize pelo menos uma diretiva de compilação e uma classe de armazenado

```
int maxProfit(int precos[], int  
bonus[], int n)
```

Exemplo:

Entrada:

```
int precos[] = {7, 1, 5, 3, 6, 4};  
int bonus[] = {0, 0, 1, 0, 2, 0};
```

Saída (em tela):

Dia de Compra: 1

Dia de Venda: 4

O Lucro Máximo é: 7

16/10/23

1) #include <stdio.h>

~~int main()~~

3,0

```
int findsum(int a[], int n){
    int soma, all;
    if (n == 1){
        soma = a[n-1];
        all = findsum(a, n-1);
        soma = a[n-1] + all;
    }
    return soma;
}
```

está fora do bloco  
do else?

```
int main(){
    int arr[5] = {-1, -2, -3, -4, -5};
    int n = 5;
    int sum = findsum(arr, n);
    printf("a soma dos valores da array: %d \n", sum);
    return 0;
}
```

2) #include <stdio.h>

```
int maxprofit (int prices[], int bands[], int n) {
```

```
// procura a menor valor de compra e a maior para a venda
```

```
int i, menor, maior, j, k, attk, L, dif, bands_m;
```

```
for (i=0; i<n; i++) {
```

```
    menor = 1000;
```

```
    if (prices[i] < menor)
```

```
        menor = prices[i];
```

```
    j = i;
```

```
    for (k=0; j < k || k < n; k++) {
```

```
        attk = 1000; attk = 0;
```

```
        if (prices[k] > attk) {
```

```
            attk = prices[k];
```

```
            L = k;
```

```
        }
```

```
        dif = attk - menor;
```

```
        bands_m = bands[L];
```

```
    }
```

```
    maior = dif + bands_m;
```

```
    if (maior < 0)
```

```
        return 0, j, L;
```

```
    else
```

```
        return maior, j, L;
```

```
int main() {
```

```
    int prices[6] = {7, 1, 5, 3, 6, 4};
```

```
    int bands[6] = {0, 0, 1, 0, 2, 0};
```

```
    int n = 6;
```

```
    int lucro, compra, venda = maxprofit(prices, bands, n);
```

```
    printf("dia de compra: %d \n", compra);
```

```
    printf("dia de venda: %d \n", venda);
```

```
    printf("o lucro máximo é: %d \n", lucro);
```

```
    return 0;
```

do próprio array

→ (lucro de arrm.?)

algum valor?

fora do bloco?

loop infinito?

não tá sendo incrementado

→ acertar na próxima

// onde a animação, precisa ter atenção nas condições para  
entender onde acaba cada loop.

15

que notação é essa?