

Prova 1

Algoritmos e Programação 2

2^o semestre de 2022

1 Prova

1. Considerando a função a seguir, quantas chamadas são realizadas à função HH (incluindo a primeira chamada externa da função), se o valor de $m = 5$ e o valor de $n = 2$? Desenhe as chamadas para deixar registrado suas contas.

```
int HH(int m, int n)
{
    if (n == 1)
        return m+1;
    else if (m == 1)
        return n+1;
    else
        return HH(m, n-1) + HH(m-1, n);
}
```

2. O esqueleto do programa contido no arquivo **produtos.cpp** (armazenado dentro das pasta Documentos) auxilia no gerenciamento dos produtos vendidos pelo site **vendiAll**. O programa já contém:
 - A definição de um registro contendo quatro campos para representar os dados de um produto: nome do produto (tamanho máximo 20), preço, data de vencimento e quantidade em estoque. Além de um corpo incompleto da função **main** e outras funções a serem criadas.Escreva um programa que cadastre **n** produtos (nome, preço, quantidade em estoque e data de vencimento), sendo **n** informado pelo usuário (**n** \leq 20). Em seguida, imprima uma lista com o nome e data de vencimento dos produtos que já atingiram a data de vencimento (a data a ser consultada como data de vencimento é uma data **dt** informada pelo usuário). Para isso, crie uma função com o protótipo a seguir:

```
void listaVencidos(tipoData dt, int n, tipoProd v[]);
```

O programa deve ainda, para aqueles produtos que estão com estoque abaixo do mínimo (inclusive o mínimo que é de 5 unidades), calcular e determinar qual o produto com maior valor final de venda (qtde*preço) do produto. Para isso, crie uma função que verifique quem é o produto com maior valor final de venda e retorne o índice onde ele se encontra (suponha que não há empate). Os dados (nome, maior valor final e quantidade em estoque) devem ser impressos na **main**. Se não houver produtos com estoque mínimo, imprima a mensagem "Produtos inexistentes".

int estoqueMin(int n, tipoProd v[]);

3. Assinale as verdadeiras.

Escolha uma ou mais:

- (a) Pode-se usar o nome de um vetor V para acessar indiretamente o conteúdo do elemento na primeira posição do vetor, ou seja, $*V == V[0]$.
 - (b) Supondo que exista um vetor de inteiros $V = 1,2,3,4,5$ e que a declaração de um ponteiro p ($\text{int } *$) tenha sido realizada. Se a instrução $p = V+(n-1)$ for executada, o valor de $*(p-2)$ será 3.
 - (c) Supondo que exista um vetor inteiro $V = 1,2,3,4,5$ e que a declaração de um ponteiro p ($\text{int } *$) tenha sido realizada. Se a instrução $p = V + 1$ for executada, o valor de $p[2]$ será 2.
 - (d) Se V é o nome de um vetor de tamanho n , então $V+n$ é o endereço da última posição do vetor.
 - (e) Pode-se usar o nome de um vetor V para acessar diretamente o conteúdo do elemento na primeira posição do vetor, ou seja, V é equivalente a $V[0]$.
4. Suponha um programa que leia n números inteiros, armazenando-os em um vetor, de tamanho máximo 30, calcule e imprime se o vetor possui um número primo. Para isso, crie uma função **recursiva temPrimo**, que recebe um vetor com n elementos e retorne **true** se o vetor possui um número primo, **false** em caso contrário.
5. Uma sub-rotina, escrita numa linguagem de programação que chama a si mesma, direta ou indiretamente, ie dita I . O uso de II geralmente permite uma descrição mais clara e concisa dos algoritmos, especialmente quando o problema tem esta característica por natureza, como é o caso III , por exemplo. Um compilador implementa este tipo de sub-rotina por meio de uma IV , na qual são armazenados os dados usados em cada chamada da sub-rotina que ainda não terminou de processar.
- As lacunas são corretas e respectivamente, preenchidas por:

- (a) I - autoexecutável, II - auto execução, III - das árvores, IV - pilha encadeada.
 - (b) I - recursiva, II - recursividade, III - do fatorial, IV - pilha
 - (c) I - loop, II - looping, III - da pesquisa binária, IV - árvore de decisão
 - (d) I - recursiva, II - recursão, III - dos vetores, IV - fila
 - (e) I - loop, II - looping, III - da série de Fibonacci binária, IV - árvore binária
6. Qual será o valor da expressão (x+y) ao final do trecho de código abaixo? Antes de apresentar a resposta final da questão, mostre também os valores de x e de y.
- (a) `int x, y, *p;`
 - (b) `y = 0;`
 - (c) `p = &y;`
 - (d) `x = *p;`
 - (e) `x = 4;`
 - (f) `(*p)++;`
 - (g) `-x;`
 - (h) `(*p) = (*p) + x;`

2 Gabarito

1. Resposta: 9, é necessário desenhar as chamadas para deixar registrado as contas.!!
2. Reposta:

```
#include <stdio.h>
#define MAX 20

struct tipoData
{
    int d, m, a;    // dia, mes, ano
};

struct tipoProd
{
    char nome[MAX+1];
    double preco;
    tipoData dtvenc;
```



```

int estoqueMin(int n, tipoProd v[])
{
    int indice = -1;
    double maiorValor = -1;
    for(int i=0; i<n; i++)
    {
        if(v[i].estoque <= 5)
        {
            double valor;
            valor = (v[i].estoque * v[i]

            if (valor>maiorValor)
            {
                maiorValor = valor;
                indice = i;
            }
        }
    }
    return indice;
}

```

3. Verdadeiras: A, B

4. Resposta:

```

#include <stdio.h>
int verificaPrimo(int w){
    int i = 0;
    if(w==2){
        return 1;
    } else if (w==1){
        return 0;
    }
    for(i = 2; i < w && w%i!=0; i++)
    ;
    if(i==w)
    return 1;
    else
    return 0;
}
int temPrimo(int v[] , int k){
    if(verificaPrimo(v[k-1])){
        return 1;
    } else{
        if(k==0){
            return 0;
        }
    }
}

```

```

        }else{
            return temPrimo(v, k-1);
        }
    }
}
int main(){

    int n;
    scanf("%d", &n);

    int v[n];
    for(int i=0; i<n; i++){
        scanf("%d", &v[i]);
    }
    if(temPrimo(v, n) == 1)
        printf("Sim\n");
    else
        printf("Nao\n");

    return 0;
}

```

5. Verdadeira: B
6. $x = 3, y = 4, x+y = 7$