

Rio de Janeiro, 27/03/2020

Aluno: João Pedro Espechit Silveira – 2019200901

Professora: Priscilla Fonseca de Abreu Braz

AV1:

Q1.a: Uma função recursiva se caracteriza pelo fato da própria realizar a chamada de si mesma por meio do comando return no final da sua execução.

Q1.b: A função recursiva funciona de forma que sempre que ela for chamada dentro do programa, o sistema reserva memória para as variáveis e parâmetros dessa função. A função no topo da pilha é a que está sendo executada e quando uma função termina, ela é removida da pilha

Q1.c: Ela tem como vantagens a capacidade de simplificar a solução de alguns problemas, possibilitando também que o código fique mais compacto e esteticamente mais agradável. Porém, são, de forma geral, mais lentas e consomem mais recursos da máquina, se comparadas com outras soluções.

Um exemplo prático para o uso de funções recursivas pode ser visto para calcular números fatoriais. É possível fazer esse cálculo tanto por meio de um laço for

```
for(counter=number;counter>0;counter--)  
    fat=fat*counter;  
//considerando que fat foi inicializada valendo 1
```

Ou em uma função recursiva

```
if(number<=1)  
    return 1;  
else  
    return fFat(number-1)*n;  
// em que fFat é o nome da função
```

Q1.d:

- (1) – int f(int n){ //Declaração da função f;
- (2) – if (n == 0)
- (3) – return 0; //condição 1, a função retorna 0, se n=0 (linhas 2 e 3);
- (4) – if (n == 1)
- (5) – return 1; //condição 2, a função retorna 1, se n=1 (linhas 3 e 4);

(6) – else

(7) – `return(f(n-1)+ 3 * f(n-2));` // condição 3, é feito o cálculo da função recursiva quando n não atende as condições anteriores

Rio de Janeiro, 27/03/20

Aluno: João Pedro Espedit Silveira - 2019200901

Professora: Priscilla Fereira de Abreu Braz

AVI

Q1a Uma função recursiva se caracteriza pelo fato de a própria realizar a chamada de si mesma por meio do comando return no final de sua execução

Q1b A função recursiva funciona de forma que sempre que ela for chamada dentro do programa, o sistema reserva memória para as variáveis e parâmetros dessa função. A função no topo da pilha é a que está sendo executada e quando uma função termina, ela é removida da pilha.

Q1c Ela tem como vantagens a capacidade de simplificar a solução de alguns problemas, possibilitando também que o código fique mais compacto e esteticamente agradável. Porém, são, de forma geral, mais lentas e consomem mais recursos da máquina, se comparadas com outras soluções.

Um exemplo prático para o uso de funções recursivas pode ser visto para calcular números fatoriais. É possível fazer esse cálculo tanto por meio de um laço for

```
for(counter = number; counter >= 1; counter--)
```

```
    fat = fat * counter;
```

// Considerando que fat foi inicializada valendo 1

Ou em uma função recursiva

↳

```
if (number <= 1)
```

```
    return 1;
```

```
else
```

```
    return fFib(number-1)*n;
```

// em que fFib é o nome da função

Q1. di int f(int n) { // declaração da função f }

```
    if (n == 0)
```

```
        return 0; // condição 1, a função retorna 0 se n=0
```

```
    if (n == 1)
```

```
        return 1; // condição 2, a função retorna 1, se n=1
```

```
    else
```

```
        return (f(n-1) + 3*f(n-2));
```

// condição 3, é feito o cálculo da função recursiva quando n não atende as condições anteriores